



# NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

## THESIS

**IMPROVED USABILITY OF LOCOMOTION DEVICES  
USING HUMAN-CENTRIC TAXONOMY**

by

Alex Mabini

March 2009

Thesis Advisor:  
Second Reader:

Rudolph Darken  
Joe Sullivan

**Approved for public release; distribution is unlimited**

THIS PAGE INTENTIONALLY LEFT BLANK

**REPORT DOCUMENTATION PAGE**

Form Approved OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

<b>1. AGENCY USE ONLY (Leave blank)</b>	<b>2. REPORT DATE</b> March 2009	<b>3. REPORT TYPE AND DATES COVERED</b> Master's Thesis	
<b>4. TITLE AND SUBTITLE:</b> Improved Usability of Locomotion Devices Using Human-Centric Taxonomy		<b>5. FUNDING NUMBERS</b>	
<b>6. AUTHOR(S)</b> Mabini, Alex		<b>8. PERFORMING ORGANIZATION REPORT NUMBER</b>	
<b>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</b> Naval Postgraduate School Monterey, CA 93943-5000		<b>10. SPONSORING/MONITORING AGENCY REPORT NUMBER</b>	
<b>9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES)</b> N/A			
<b>11. SUPPLEMENTARY NOTES</b> The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
<b>12a. DISTRIBUTION / AVAILABILITY STATEMENT</b> Approved for public release; distribution is unlimited		<b>12b. DISTRIBUTION CODE</b>	
<b>13. ABSTRACT (maximum 200 words)</b>  This thesis investigates how early taxonomies of locomotion fail to provide a comprehensive enough framework to facilitate usable locomotion devices due to a failure of understanding the human component in interaction. It then proposes an alternative human-centric taxonomy for locomotion that grounds itself on the physiological, physical and extra-physical cues the human body is capable of providing rather than only the input existing interaction devices are capable of receiving. Through the realization that interaction begins with the human, not the machine, this thesis is able to determine a cue from the body that is able to provide enough information for use by an algorithm to recognize walking and running forward, sidestepping, back stepping, and jumping with a minimal amount of sensors and associated hardware. This thesis then develops and performs initial tests on a fully implemented locomotion device using input from two inertial sensors on the legs in conjunction with the locomotion recognition algorithm for use in any commercial-off-the-shelf (COTS) video game for PCs that use keypresses for locomotion input.			
<b>14. SUBJECT TERMS</b> Locomotion, Virtual Walking, Taxonomy, Virtual Environments			<b>15. NUMBER OF PAGES</b> 121
			<b>16. PRICE CODE</b>
<b>17. SECURITY CLASSIFICATION OF REPORT</b> Unclassified	<b>18. SECURITY CLASSIFICATION OF THIS PAGE</b> Unclassified	<b>19. SECURITY CLASSIFICATION OF ABSTRACT</b> Unclassified	<b>20. LIMITATION OF ABSTRACT</b> UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)  
Prescribed by ANSI Std. Z39-18

THIS PAGE INTENTIONALLY LEFT BLANK

**Approved for public release; distribution is unlimited**

**IMPROVED USABILITY OF LOCOMOTION DEVICES USING HUMAN-CENTRIC TAXONOMY**

Alex T. Mabini  
Lieutenant Commander, United States Navy  
B.S., United States Naval Academy, 1997

Submitted in partial fulfillment of the  
requirements for the degree of

**MASTER OF SCIENCE IN MODELING, VIRTUAL ENVIRONMENTS AND  
SIMULATION (MOVES)**

from the

**NAVAL POSTGRADUATE SCHOOL**  
**March 2009**

Author: Alex T. Mabini

Approved by: Rudolph Darken  
Thesis Advisor

CDR Joseph Sullivan  
Second Reader

Mathias Kolsch  
Chair, MOVES Academic Committee

THIS PAGE INTENTIONALLY LEFT BLANK

## **ABSTRACT**

This thesis investigates the failure of early taxonomies for locomotion to provide a comprehensive enough framework in facilitating the development of usable locomotion devices due to an inadequate classification of the human component. It then proposes an alternative human-centric taxonomy for locomotion that grounds itself on the physiological, physical and extra-physical cues the human body is capable of providing rather than only the input existing interaction devices are capable of receiving. Through the realization that interaction begins with the human, not the machine, this thesis is able to determine a cue from the body that is able to provide enough information for use by an algorithm to recognize walking and running forward, sidestepping, back stepping, and jumping with a minimal amount of input. This thesis then develops and performs initial tests on a fully implemented locomotion device using input from two inertial sensors on the legs in conjunction with the locomotion recognition algorithm for use in any commercial-off-the-shelf (COTS) video game for PCs that use keypresses for locomotion input.

THIS PAGE INTENTIONALLY LEFT BLANK

## TABLE OF CONTENTS

I.	INTRODUCTION.....	1
A.	PROBLEM STATEMENT.....	1
B.	MOTIVATION.....	2
C.	THESIS GOALS.....	3
D.	THESIS ORGANIZATION.....	3
II.	ANALYSIS OF PAST TAXONOMIES .....	5
A.	INTRODUCTION.....	5
B.	PAUL FITTS, 1954.....	5
1.	Details.....	5
2.	Significance.....	6
C.	BOWMAN, KOLLER AND HODGES, 1997.....	8
1.	Details.....	8
2.	Significance.....	10
D.	ARNS, 2002 .....	11
1.	Detail.....	11
2.	Significance.....	12
E.	SUMMARY .....	13
III.	HUMAN-CENTRIC TAXONOMY OF INTERACTION DEVICES.....	15
A.	INTRODUCTION .....	15
B.	THEORY OF INTERACTION .....	16
C.	ORDER OF IMMEDIACY, [OI] .....	17
1.	Assumptive Model for Order of Immediacy .....	18
2.	First Order — Physiological.....	19
3.	Second Order — Intra-Physical .....	21
4.	Third Order — Inter-Physical .....	22
5.	Fourth Order — Extra-Physical .....	23
6.	Implications.....	25
D.	DEGREES OF INPUT, [DI] .....	25
1.	Physiological Inputs .....	26
2.	Physical Inputs .....	26
3.	Extra-Physical Inputs .....	28
E.	DEGREES OF OUTPUT, [DO] .....	29
F.	OVERALL TAXONOMY.....	31
1.	Metrics .....	33
a.	<i>Order of Immediacy, [OI]</i> .....	33
b.	<i>Ratio of Output to Input, [DI:DO]</i> .....	33
c.	<i>Degrees of Input Conserved, [DI<sup>C</sup>]</i> .....	34
G.	NAMING INTERFACES USING PROPOSED TAXONOMY .....	34
1.	Suggested Naming Convention.....	34
2.	Suggested Abbreviation Rules.....	34
3.	Examples .....	35

<b>IV.</b>	<b>IMPLEMENTATION.....</b>	<b>37</b>
A.	DETERMINATION OF A CUE .....	37
B.	2 <sup>ND</sup> ORDER, 2 LEG LOCOMOTION RECOGNITION ALGORITHM .	37
1.	General Design .....	37
2.	Internal Structure.....	39
3.	The Payload.....	41
C.	ALGORITHM STEPS .....	41
1.	Analyze Leg Positions.....	42
a.	<i>Identify Left Leg Position</i> .....	42
b.	<i>Identify Right Leg Position</i> .....	43
c.	<i>Identify Leg Position Situation Satisfied by the Combination of Left and Right Leg Positions</i> .....	44
2.	Analyze Leg Velocities .....	45
a.	<i>Identify Velocity of Left Leg</i> .....	45
b.	<i>Identify Velocity of Right Leg</i> .....	46
c.	<i>Identify Leg Velocity Situation (VS) from One of 17 Situations using Velocity Matrix (VM)</i> .....	47
3.	Identify New Movement Type.....	47
4.	Check and Set New Movement State .....	48
5.	Use Conditionals to Determine and Set New Movement Speed .....	48
D.	ILLUSTRATION OF ALGORITHM USAGE.....	48
1.	Upswing Phase .....	48
2.	Downswing Phase .....	50
E.	CONCLUSION .....	50
<b>V.</b>	<b>PRELIMINARY RESULTS.....</b>	<b>53</b>
A.	HUMAN ABILITY REQUIREMENTS ANALYSIS USING F-JAS.....	54
1.	Accelerate from Rest to Walk or Jog .....	57
2.	Decelerate from Walk or Jog to Rest .....	57
3.	Accelerate from Walk to Jog.....	58
4.	Decelerate to Walk from Jog.....	58
5.	Walk .....	58
6.	Jog .....	58
7.	Turn in Place (no forward or side movement).....	58
8.	Sidestep.....	59
9.	Tilt Upper-Body (without foot movement) .....	59
B.	TEST RESULTS.....	59
C.	CONCLUSION .....	60
<b>VI.</b>	<b>CONCLUSION AND FUTURE WORK.....</b>	<b>61</b>
A.	CONCLUSION .....	61
B.	FUTURE WORK.....	61
<b>APPENDIX A.</b>	<b>TRANSITION MATRICES .....</b>	<b>63</b>
<b>APPENDIX B.</b>	<b>TRANSITION DIAGRAM NOTES .....</b>	<b>69</b>

<b>APPENDIX C. C++ IMPLEMENTATION OF ALGORITHM .....</b>	<b>83</b>
<b>INITIAL DISTRIBUTION LIST .....</b>	<b>103</b>

THIS PAGE INTENTIONALLY LEFT BLANK

## LIST OF FIGURES

Figure 1.	Fitts' Law .....	5
Figure 2.	Accot-Zhai Steering Law .....	7
Figure 3.	Bowman's Taxonomy of Locomotion.....	9
Figure 4.	Arns's Taxonomy of Locomotion .....	12
Figure 5.	Model of human interaction adaptation from David Meister's, <i>Human Factors: Theory and Practice</i> , 1971 .....	16
Figure 6.	Orders of immediacy .....	19
Figure 7.	Electrochemical excitation of neurons .....	20
Figure 8.	The motor tract .....	21
Figure 9.	Function of antagonistic pairs in muscles .....	22
Figure 10.	Physical interaction between right leg and ground .....	24
Figure 11.	Bones of the hand .....	27
Figure 12.	Fully articulated limb segments of the body.....	28
Figure 13.	Proposed taxonomy of human interaction .....	32
Figure 14.	Hick's law .....	33
Figure 15.	Placement of leg sensors .....	38
Figure 16.	22-LRA model .....	39
Figure 17.	Example of state-transition diagram for auction process .....	40
Figure 18.	Overall state-transition diagram.....	40
Figure 19.	Graph describing how the position of each leg is determined by roll and pitch.....	43
Figure 20.	Illustration of frontal (coronal) and sagittal planes .....	45
Figure 21.	Graph describing the relationship between pitch and roll velocity .....	46
Figure 22.	Illustration of implemented demo in use .....	53

THIS PAGE INTENTIONALLY LEFT BLANK

## LIST OF TABLES

Table 1.	Position situation matrix.....	44
Table 2.	Velocity situation matrix.....	47
Table 3.	Transition matrix for stand state .....	48
Table 4.	Transition matrix for forward upswing and downswing states.....	49
Table 5.	Absense/presence analysis of natural locomotion.....	54
Table 6.	Scaled analysis of natural locomotion .....	55
Table 7.	Absense/presence analysis of VE locomotion on the ODT .....	55
Table 8.	Scaled analysis of VE locomotion on the ODT .....	56
Table 9.	Absence/presence analysis of LocoX using implementation .....	56
Table 10.	Scaled analysis of LocoX using implementation.....	57

THIS PAGE INTENTIONALLY LEFT BLANK

## **ACKNOWLEDGMENTS**

I would like to express my utmost gratitude to Dr. Rudy Darken and CDR Joe Sullivan for their extreme patience in this work. They provided me with inspiration both academically as well as professionally while at the Naval Postgraduate School. I would also like to thank CAPT Ricks Polk, CAPT Michael Slotsky, and CAPT David Taylor for their firm leadership and for helping facilitate the completion of this work while at sea.

THIS PAGE INTENTIONALLY LEFT BLANK

## I. INTRODUCTION

### A. PROBLEM STATEMENT

Virtual environments (VEs) have seen extraordinary advances since the early days of monochrome graphics and wireframe models. Today's video cards are able to produce three-dimensional environments in real time with ever increasing photorealism. Furthermore, developments in audio technology continue to increase the spatial awareness of users in VEs. Once only available within dedicated simulators, the VEs of today can be found everywhere, fully networked, so that experiences can be shared with thousands of users around the world. However, despite the extraordinary advances in VEs, little improvement has been made in how naturally users interact with these environments. Amongst all interaction methods, locomotion, the movement of a user's viewpoint through a VE, is arguably the most important, yet has seen the least amount of progress and remains one of the most perplexing issues facing designers of VEs—how to travel through unlimited virtual space using natural walking movements while remaining within the constraints of limited physical space. The fictitious Holodeck of Star Trek provides an idealization of the perfect locomotion device, but its realization remains elusive. Instead, the most widely used interaction device for virtual environments remains the mouse and keyboard.

I hypothesize that locomotion devices have seen such little improvement compared to other VE technologies, as well as other interaction methods, due to a lack of understanding of the cues they rely on and how those cues influence the resulting characteristics of the device. Without such an understanding, design of locomotion devices cannot move forward purposefully. I contend that by using a human-centric taxonomy rather than the device-centric taxonomies of the past, it is possible to provide the framework and language necessary to

facilitate the improved usability of not only locomotion devices, but interaction devices of all manners, providing insight into usability limitations simply by their location in the taxonomic structure.

## B. MOTIVATION

While studies continue to develop locomotion devices, many do so haphazardly, designing marvels of mechanical science with fundamental flaws in usability. Additionally, for many tests, researchers use different standards for measurement. As a result, designers of locomotion devices are left with little guidance, resulting in novel devices that solve certain aspects of locomotion, but then create problems for other aspects. If we assume the International Standards Organization (ISO) 16982 definition of usability as “the effectiveness, efficiency and satisfaction with which a specified set of users can achieve a set of tasks in a particular environment”, it can be argued that the most usable locomotion device remains the mouse and keyboard, evidenced by the millions of satisfied users across the world logging millions of hours playing immersive, massively multi-player online first-person shooter (MMOFPS) video games. No other device offers as many functions in as well a recognized form as the mouse and keyboard. Despite this, the mouse and keyboard implementation remains far from ideal. Although the overall form of the device may be well-recognized, the mapping of keys to resulting functions is by no means intuitive, and must be learned and relearned after periods of non-use. Additionally, the number of keys the user’s fingers can reach without forcing the user to look down at his hand limits the number of functions the user can easily access. Even so, current locomotion devices lack the efficiency of the mouse and keyboard due to their enormous footprints and associated costs with no significant increase in functionality. For military use, locomotion devices have failed, and will continue to fail in meeting usability requirements until, for example, they can be used by Marines en route Afghanistan, networked wirelessly within the confined space of

an amphibious assault ship hangar bay, rehearsing squad tactics in virtual maps of the remote, mountainous frontier of the Afghanistan-Pakistan border.

### C. THESIS GOALS

The overall goals of this thesis are:

- To develop taxonomy for locomotion devices centered on the possible cues provided by the human body.
- To determine the optimal cue from natural or metaphoric bipedal locomotion upon which to base a locomotion device on with the goal of superseding the mouse and keyboard in terms of overall usability.
- To develop an algorithm which recognizes locomotion based on the determined optimal cue
- To develop a working locomotion device utilizing the locomotion recognition algorithm

### D. THESIS ORGANIZATION

Six chapters comprise this research:

- *Chapter I* — Introduction. Identifies the purpose, motivation and goals of this research.
- *Chapter II* — Analysis of Past Taxonomies. Analyzes past models of interaction and locomotion taxonomies, explaining their significance in the development of usable interaction devices.
- *Chapter III* — Human-Centric Taxonomy of Interaction Devices. Proposes an alternative, human-centric taxonomy of interaction devices based on the possible cues provided by the human body. Explains where locomotion devices fit into this model.
- *Chapter IV* — Implementation. Discusses research into natural, bipedal locomotion and the use of the proposed human-centric taxonomy in search of an optimal cue upon which to base a locomotion device on. Using the optimal cue, it then discusses a fully implemented locomotion device that recognizes natural, walk-in-place motions as metaphors for locomotion.

- *Chapter V* — Preliminary Results. Conducts human ability requirements (HAR) analysis of implementation and explains initial test results
- *Chapter VI* — Conclusions and Future Work. Discusses work to be done to further test or validate both the taxonomy as well as the implementation of the locomotion device.

## II. ANALYSIS OF PAST TAXONOMIES

### A. INTRODUCTION

This chapter discusses past models of human movement and taxonomies of locomotion, analyzing their significance as well as their shortcomings in the development of usable interaction devices. I begin with an explanation of Fitts' law, followed by an examination of two taxonomies of locomotion.

### B. PAUL FITTS, 1954

In 1954, Paul Fitts published an article revealing a model of human motor performance which predicts, to an unusually accurate degree, the time required to rapidly point to a target area with various devices based on the distance to the target and the width of the target area [1]. Known as Fitts' law, this model contributed to the introduction of the modern mouse, which has remained the most commonly used human-computer interaction device since Apple's introduction of the Macintosh in 1984.

$$T = a + b \log_2 \frac{D}{W}.$$

Figure 1. Fitts' Law

#### 1. Details

In his experiments, Fitts had subjects move a stylus back and forth between two target areas on a desk surface in a minimum amount of time. The results of his experiments closely approximated the formula shown in Figure 1. The logarithm in the formula represents the index of difficulty of the target, and depends on the ratio of the distance (D) from the center of the target to the width (W) of the target area. It can also be looked at in terms of angular width if we recognize the function within the logarithm to be the cotangent of the angular width.

If we assume a constant distance to a target and decrease the width of the target area, the accuracy required to successfully arrive at a point within the smaller target area increases, and, according to Fitts, the speed of subjects who successfully do so decreases. Essentially, Fitts' law establishes a logarithmic trade-off between speed and accuracy in rapid, spatially constrained limb movements [2].

Thousands of tests, under varying conditions with different limbs and with users of varying conditions (young, old, drugged, etc.) all seem to confirm the relationship Fitts' law models [3]. Plotted results from experiments using Fitts' law typically fall along a straight line with a correlation greater than 0.95.

## 2. Significance

The insight Fitts' law provides is of landmark significance to the development of interaction devices. In 1978, Card, English, and Burr conducted a study comparing the performance of different input devices, including the mouse, as human-computer interaction devices. In their experiments, the mouse provided the fastest and most accurate input, which led to the mouse's commercial introduction, and has been, alongside the keyboard, the staple of human-computer interaction for over a quarter of a century [4].

Another significant aspect of Fitts' law is that the movement which it explains—rapid, spatially constrained movement—serves as the building block for countless everyday activities such as pointing, looking, and most relevant to this work, walking [3]. It models behavior at so low a level that whether the movement is the result of trained behavior or not, it remains successful in its prediction [3]. In addition to its ability to model behavior at the lowest level, it may be possible to model complex motions as well by reducing the articulated movement of our limbs to a combined series of rapid, aimed movements, somewhat similar to the Accot-Zhai steering law [5], which provides extremely accurate predictions for two-dimensional movement, shown in Figure 2.

$$T = a + b \int_C \frac{ds}{W(s)}$$

Figure 2. Accot-Zhai Steering Law

Fitts' law also ties cognitive psychology to motor performance. Various models explaining how Fitts' law works, such as the deterministic iterative-corrections model and the more recent stochastic optimized-submovement model, link the cognitive processes of target selection and mid-movement adjustment to motor performance [3]. In experiments with the stochastic model, Meyer et al. identified acceleration changes within rapid, aimed movements marking the beginning and end of submovements. The stochastic nature of these submovements demonstrates the limitations of human motor systems and explains the trade off between speed and accuracy, which we see in Fitts' law, as the psychophysical attempt to cope with neuromotor noise [3]. The degree to which we observe this trade off may indicate the relative ease-of-use or difficulty a particular interaction entails [2].

Perhaps the most significant aspect of Fitts' law, however, is how it explains the effect of constrained dimensions on a device's index of difficulty (ID). For example, in a typical window based graphics user interface, the borders constrain the horizontal and vertical movement of a mouse cursor such that buttons in the corners essentially have the equivalent of infinite widths, effectively making the ID of the device nearly nonexistent for that particular movement, thus allowing high speed interactions. Another example involves typing on a keyboard. If we are to consider each keystroke as a rapid, aimed movement, we see that movement is constrained by the mechanical design of the key such that it prevents lateral movement and minimizes the vertical travel required to complete a keystroke, thereby creating the equivalent of infinite target widths in every dimension. This particular example of a rapid, spatially constrained movement in an interaction device, results in approximately 4 characters per second for the average skilled typist. Essentially, creating border constraints

limits movement in the dimension perpendicular to the border, thereby creating an infinite target width along the constrained dimension beyond the border. This effectively decreases a device's ID, requiring less accuracy to perform a function at higher speeds.

Although Fitts' law explains to an extremely accurate degree rapid, aimed movement, which forms the most basic building blocks of human interaction, the measures of performance it predicts, namely speed and accuracy, do not necessarily explain all the qualities that make a device usable. Despite this shortcoming, the robustness and flexibility of Fitts' law make it the cornerstone of all human interaction predictive models, and, as we shall see, will prove useful in developing a cue-based taxonomy for human interaction.

### **C. BOWMAN, KOLLER AND HODGES, 1997**

In 1997, Doug Bowman, David Koller, and Larry Hodges of the Georgia Institute of Technology published the first cohesive taxonomy describing locomotion in VEs. This taxonomy paved the way for a more thorough understanding of locomotion devices and provided a set of metrics beyond that which Fitts' law predicted.

#### **1. Details**

Bowman et al. began their taxonomy of locomotion by defining travel in a VE as control of a user's viewpoint motion. They then identified three basic characteristics of locomotion: (1) direction and target selection, (2) velocity and acceleration selection, and (3) input conditions. They further classified each of these characteristics as shown in Figure 3.

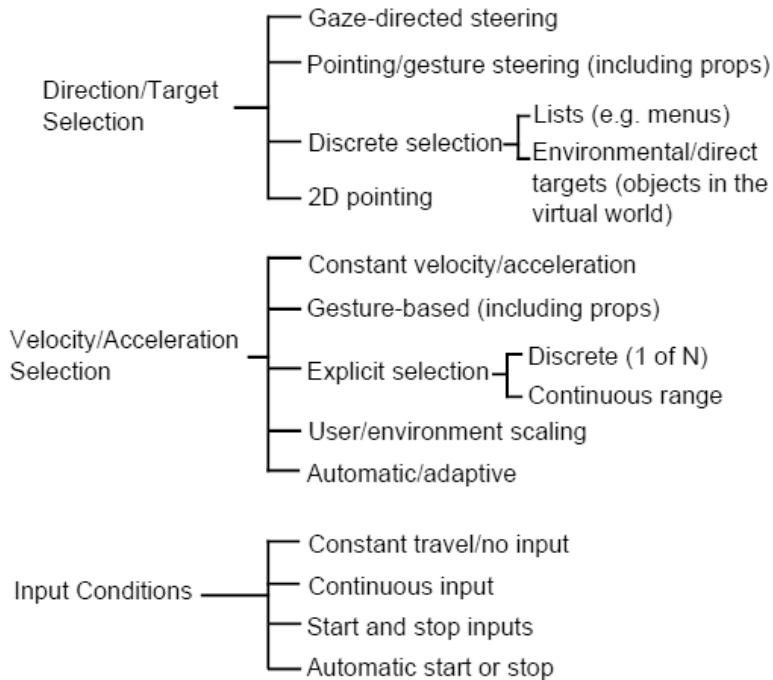


Figure 3. Bowman's Taxonomy of Locomotion

Recognizing the necessity of defining qualities other than only speed and accuracy as measures for the effectiveness of locomotion, Bowman et al. added spatial awareness, ease of learning, ease of use, information gathering, and presence to the list of qualities effective travel techniques promote.

With their model for locomotion defined, Bowman et al. then proceeded to compare two direction selection techniques: gaze-directed steering and pointing. With gaze-directed steering, travel is in the direction the user's head is turned. With pointing, travel is in the direction the user is pointing with their hand. In their experiments, Bowman et al. determined gaze-directed steering to be faster than pointing as a travel technique, but pointing to be more accurate [6].

They also conducted an experiment to measure the disorientation caused by four velocity/acceleration travel techniques: (1) slow velocity, no acceleration, (2) fast velocity, no acceleration, (3) slow-in slow-out (SISO), and (4) jumping. In slow and fast velocity, no acceleration travel techniques, movement remains at a constant speed. In SISO travel techniques, movement begins slowly,

accelerates to a maximum speed, and finally decelerates upon reaching an intended destination. Finally, in the jumping travel technique, users teleport immediately to an intended destination. Bowman et al. determined that teleportation introduces confusion to the user, adversely affecting one of the qualities of effective travel, spatial awareness [6].

## 2. Significance

One of the significant contributions Bowman et al. made to the understanding of locomotion interaction was the introduction of additional measures of performance for effective travel techniques besides only speed and accuracy. Their measures of spatial awareness, ease of learning, ease of use, information gathering, and presence used questionnaires, which, despite being subjective, provide insight into characteristics which, as of yet, still lack purely quantitative methods of analysis. However, these characteristics are not necessarily mutually exclusive, and some may have correlation to speed and accuracy themselves.

Bowman et al. also revealed important findings through their experiments involving pointing and gaze-directed steering. In experiments validating Fitts' law, we mainly see comparisons in speed and accuracy using the same limb and similar movements varied only by the distance and width of the target. Bowman et al. on the other hand, compared the speed and accuracy of similar interactions, namely pointing, using different limbs, in this case, hands vs. the head. The speed-accuracy tradeoff exhibited between these two methods of pointing demonstrates a natural capacity for a particular limb to interact faster or more accurately in a similar movement from another limb. This provides a compelling argument for the classification of interaction devices by the limbs used in the interaction rather than by arbitrary classifications, such as "wand", or "glove", as in other taxonomies.

The main shortcoming of Bowman et al. is that although they provided a thorough model of locomotion along the lines of target selection and velocity

selection, their classification of locomotion devices solely based on input conditions misses the opportunity to tie conditions of input to the limbs used in providing the input, which their own experiments demonstrate, makes a difference in the speed and accuracy of an interaction. As a result, their taxonomy for locomotion remains device-centric and lacks the capacity to thoroughly classify locomotion devices.

#### **D. ARNS, 2002**

In her Ph.D. dissertation at Iowa State University, Laura Lynn Arns built upon the work done by Bowman, introducing her own taxonomy for locomotion techniques.

##### **1. Detail**

Arns, like Bowman, defined locomotion as the change of location of a viewpoint. However, unlike Bowman, she divided locomotion in virtual environments into two major components, rotation and translation. She then further classified each component into physical or virtual movements with the former being movement of the user relative to the virtual world and the latter being movement of the virtual world relative to the user.

After defining the components of locomotion, she then introduced a classification for interaction and display devices and showed the various ways they can be combined to create methods of locomotion. Figure 4 shows Arns' overall taxonomy.

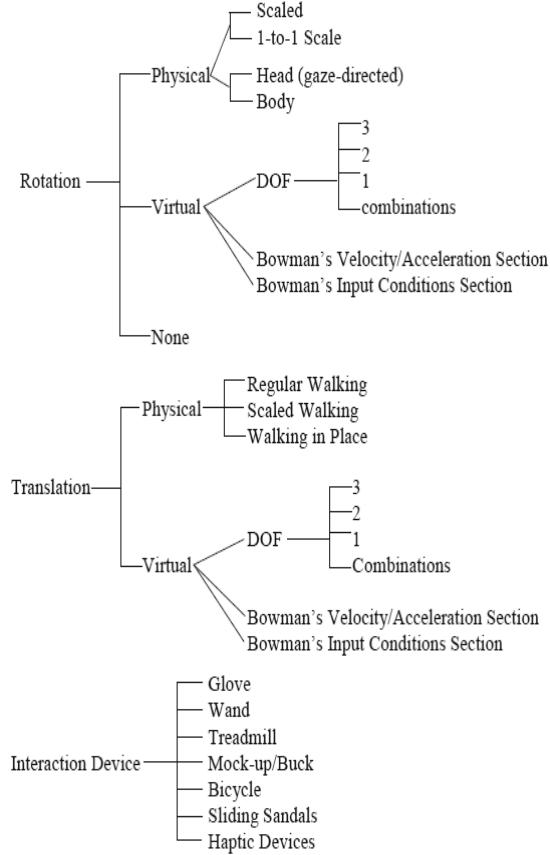


Figure 4. Arns's Taxonomy of Locomotion

## 2. Significance

The main significance of Arns' taxonomy in the development of usable locomotion devices is that, whereas Bowman et al. categorized all locomotion devices by target selection and motion towards that target, Arns' use of rotation and translation implies orientation independent of viewpoint. What this provides locomotion devices is lateral translation, as in sidestepping, which many locomotion devices lack.

Arns' taxonomy of locomotion devices, however, like Bowman's, remains device-centric, and as a result, also lacks the capacity to classify locomotion devices except through arbitrary means. Even she acknowledged the difficulty of attempting to describe and categorize all “one-of-a-kind” VR interaction devices using her proposed taxonomy [7].

## E. SUMMARY

In our analysis of Fitts' law, we learned that rapid, aimed limb movements serve as the basic building blocks of all movement. The measurable, predictable nature of these movements provides an objective way to quantitatively compare similar interactions using different limbs. Afterwards, we then discussed two taxonomies of locomotion, each of which provided thorough classifications for the components of travel. Whereas Bowman et al. divided travel by direction and speed, which is more applicable to vehicular travel, Arns divided travel into translation and rotation, introducing the ability to decouple viewpoint from direction of movement, thus providing a travel technique more applicable to natural human locomotion.

While thorough in their classifications of travel techniques, neither of the previous taxonomies places enough emphasis on the human component of interaction, instead, attempting to classify locomotion devices solely based on the function the interaction device provides or the form the interaction device takes. However, interaction is defined just as much by its input as its output, and if we wish to improve our understanding of locomotion devices through taxonomy, we require more emphasis on the human component.

THIS PAGE INTENTIONALLY LEFT BLANK

### **III. HUMAN-CENTRIC TAXONOMY OF INTERACTION DEVICES**

#### **A. INTRODUCTION**

As we have seen, past taxonomies do well to classify the components of locomotion, as well as characterize various locomotion devices using arbitrary features, however, they fail to address how the human body interacts with those devices to provide input, resulting in device-centric taxonomies. If we wish to improve the usability of locomotion devices, we must thoroughly understand how the human body interacts with those devices. This chapter proposes a taxonomy which improves upon those of the past by providing an abstract model of human interaction emphasizing the cues the human body is capable of providing rather than the types of input existing devices are capable of using. As a result, this human-centric taxonomy provides a framework for, as of yet, undeveloped locomotion devices, and provides the flexibility to describe human interaction, not only with locomotion devices, but with any human interaction device whatsoever, including keyboards, game pads, joysticks, and even devices such as cameras.

This chapter details a taxonomy for interaction devices that bases itself on various levels of cues originating from the human body. The cues, which serve as input, are then mapped to specific outputs, resulting in a comprehensive, systematic method for classifying interaction devices and techniques while providing insight into the usability of a device simply by its location within the taxonomic structure. The following list summarizes the taxa of the proposed taxonomy:

- Order of Immediacy — how “distant” a cue is from the original thought
- Degrees of Input — the signal paths providing the cue
- Degrees of Output — the functions the device provides

After defining this taxonomy, we then classify several existing locomotion devices using the proposed taxonomy and discuss how their classifications, by themselves, provide insight into the design limitations the devices either had to overcome or accept in their design.

## B. THEORY OF INTERACTION

Taxonomy is the science of classifying objects into an ordered system that indicates natural relationships. Every object has an intrinsic quality that defines it. The intrinsic quality of *living* organisms, for example, is *life*, and thus, the most recent taxonomies classify living organisms by their phylogenetic lineage. If we are to develop taxonomy for interaction devices, then, it logically follows that the intrinsic quality we must define them by is *interaction*.

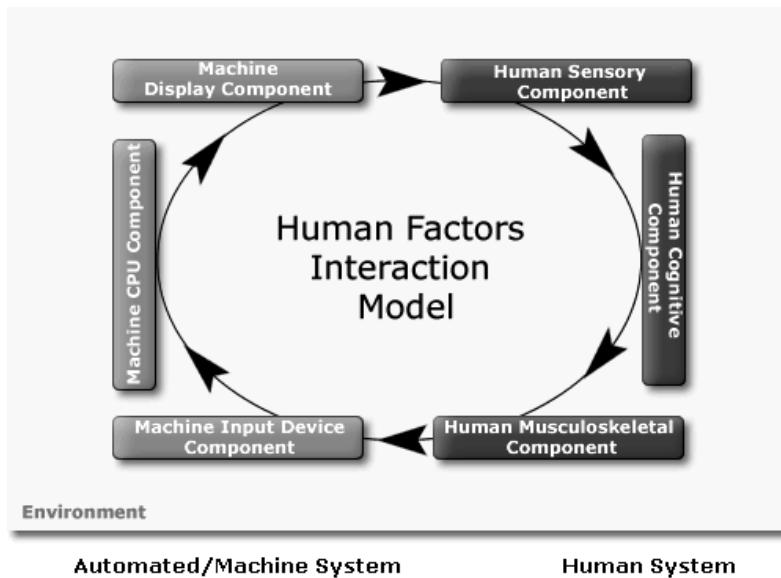


Figure 5. Model of human interaction adaptation from David Meister's, *Human Factors: Theory and Practice*, 1971

As shown in Figure 5, interaction can be defined as a mutual exchange between two objects, in the case of human factors, the “human system” and the “automated/machine system”. The human system provides input via an input device component, which, in turn processes the input via a machine CPU component, finally providing feedback to the human system through visual,

haptic, aural, or other means. Each exchange represents some type of energy conversion. The points at which the interaction occurs, where the energy is transferred between the human system and the automated/machine system, are what we recognize as the interface. If the energy transferred is coherent, the interaction becomes an exchange of information, such as a command. To illustrate this in terms of a common form of human interaction, consider that when we type, we impart energy from our fingers to the keys of a keyboard in the form of keystrokes. If the order of the keystrokes has a meaningful pattern, we may see words or sentences form on a display as feedback. This simple model of interaction, lacking in previous taxonomies, serves as the foundation for the proposed taxonomy.

### C. ORDER OF IMMEDIACY, [OI]

With the definition of interaction established, we see that the intrinsic quality of interaction originates with the human system, not within the machine input device component. It then follows that the first taxa of the proposed taxonomy should begin by classifying interaction devices by the types of input a human is capable of providing. The furthest attempt previous taxonomies make in classifying human input is through the input conditions classifier of Bowman et al. However, this classifier only addresses a single aspect of input created through physical contact with an interaction device. It does not identify the particular limb providing the input and it does not address physiological or extra-physical inputs—both being input created by something other than the human limbs, such as voice commands.

In order to create a way to capture the additional inputs available from the human body, we must examine where human input truly begins. In the example from the last section discussing interaction with a keyboard, if we consider that before a single finger ever strikes a keyboard, if not by accident, the characters, words or sentences begin as thoughts, which originate in the brain. The original thought signal becomes intent as it travels through the motor cortex of the brain.

The intent signal then passes through the brain stem, down the spinal cord, and finally to the limb intended to execute the intent. After reaching the intended limb, the signal undergoes further conversions, beginning with energy conversions at the musculoskeletal level, followed by energy conversions between the limbs of the body and an external object through physical contact, and finally, between the body as a whole and an external object using a sensor or possibly another human body in a social relationship. Each of these levels between energy conversions we identify as an order of immediacy, with the first order beginning after the initial thought becomes intent. Thus, the first taxa of the proposed taxonomy essentially describes how distant the input signal that an interaction device takes its cues from is from the original intent formed in the human brain.

### **1. Assumptive Model for Order of Immediacy**

We assume four possible energy conversions involving human interaction after the original thought signal from the brain. Thus, the assumptive model for the first level of classification is called the *Four Order Model of Immediacy*. Figure 6 illustrates the location of the signal within each order and is followed with a discussion of each order in greater detail.

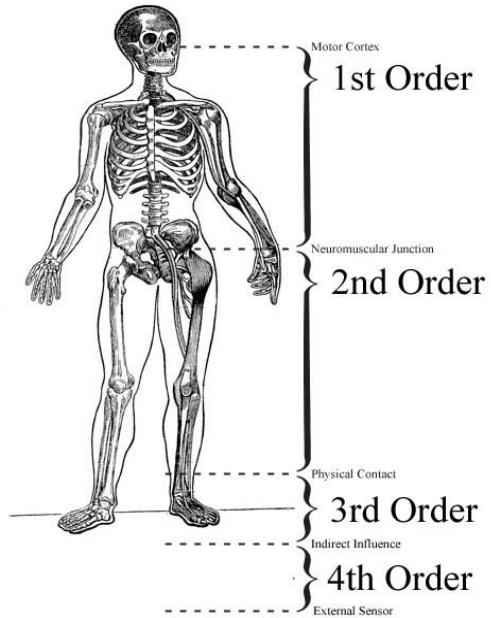


Figure 6. Orders of immediacy

## 2. First Order — Physiological

Interaction truly originates with the original thought within the brain, which manifests itself physiologically as electrochemical excitations at the synapses between adjoining neurons of the cerebral cortex [8]. These synapses excite adjacent neurons by releasing neurotransmitters, which bind to receptors in the receiving neurons, as shown in Figure 7.

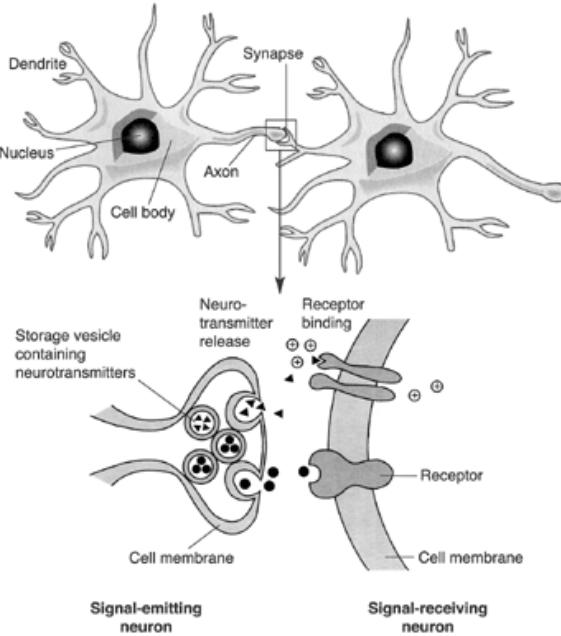


Figure 7. Electrochemical excitation of neurons

The original thought then provokes excitations within the adjoining neurons of the primary motor cortex area of the brain. This excitation of the upper motor neurons signifies the conversion of thought into intent. The conversion of the original thought signal into an intent signal marks the first significant conversion of energy and thus, the commencement of the first order of immediacy. The intent signal then travels down the corticospinal tract, through the brain stem, down the spinal cord, and finally to the muscles of the limb intended to execute the intent [9]. Figure 8 shows the motor area within the cerebral cortex and the neurological path the intent signal takes as it makes its way down the corticospinal tract to the motor unit, or the muscles.

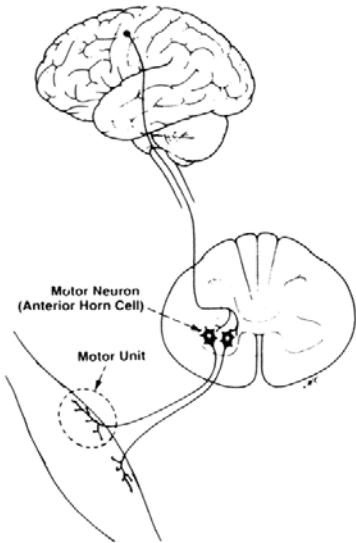


Figure 8. The motor tract

Interaction devices using the intent signal between the primary motor cortex and up to the neurological signal triggering the contraction of the muscles intended to execute the intent, but prior to the actual contraction of the muscles, rely on a physiological signal within the body, and are first order interaction devices. An example of such a device would be an electroencephalograph (EEG), which senses the electrical activity of neurons at various areas of the brain and visually displays that information, providing feedback to a patient or doctor. Treadmills that adjust speed based on heart rate also fall under this classification.

### 3. Second Order — Intra-Physical

After the intent signal travels down the motor neurons and reaches the muscles of the limb intended to perform an action, it reaches a specialized synapse known as the neuromuscular junction, which then releases neurotransmitters that trigger a reaction in the muscle fibers to which they are attached. Our muscles work mainly in antagonistic pairs, with one muscle contracting and its antagonist relaxing. This coordinated contraction and

relaxation of antagonistic muscles, known as concentric and eccentric contractions, results in the manipulation of the skeletal system of the body, as shown in Figure 9.

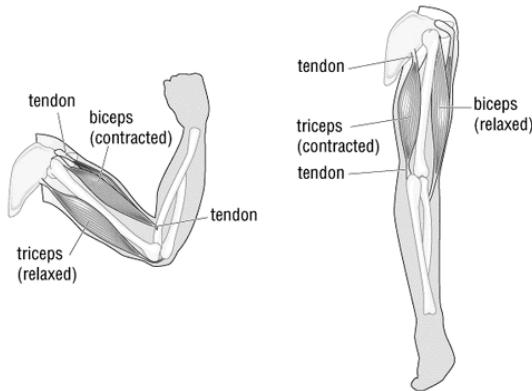


Figure 9. Function of antagonistic pairs in muscles

The manipulation of the skeletal system by the muscles signifies a conversion of electrochemical energy into physical energy, and thus, constitutes the commencement of the second order of immediacy in the signal from the original thought. Interaction devices using the contraction of the muscles and the manipulation of the body's skeletal system as a cue, rely on a physical signal within the body, and are second order interaction devices.

One key point to keep in mind is that interaction devices falling under this classification typically have sensors which are permanently in contact with the body while in operation and rely on the musculoskeletal system of the body itself as the cue, not the manipulation of an external object by the human body other than the sensor itself. An example of a second order interaction device would be a tracked head-mounted display.

#### 4. Third Order — Inter-Physical

After muscular contractions manipulate the limb intended to perform an action, that limb makes physical contact with and influences an external object. This transfer of energy between the human body and some external object through physical contact, thus, constitutes the commencement of the third order

of immediacy. Interaction devices that use the manipulation of an external object through physical contact as a cue are third order interaction devices. This classification is what most interaction devices fall under. Keyboards, mice, treadmills and touch screens all fall under this classification. The key difference between second and third order interaction devices is that the former relies on the movement of the limb itself, whereas the latter relies on the manipulation of an external object by the limb. In terms of sensor placement, the sensor for a third order interaction device typically resides in the external object, as in an optical mouse or a touch screen. However, in devices using mechanical linkages connected to the limbs, while the potentiometer sensors may not necessarily be in the linkages themselves, they sense the movement of the linkages directly, thus making them third order interaction devices. The element physical contact introduces which previous orders of immediacy do not require is ergonomics.

## **5. Fourth Order — Extra-Physical**

If an interaction device relies on a cue from some characteristic of the human body not directly caused by movement of the independently articulated limbs or an external object manipulated by those limbs, then that interaction device is a fourth order interaction device. Fourth order interaction devices, like third order interaction devices, may rely on linkages, but manipulation of those linkages are not caused by the movement of the limbs themselves. Rather, they would be manipulated by something caused by the movement of the limbs. The fourth order of immediacy signal happens to be the cue developers of most locomotion devices naturally focus on because natural bipedal locomotion itself is a fourth order interaction. For example, in natural bipedal locomotion, the interaction of our legs with the ground causes a forward displacement of our entire body. If the ground were slippery enough, the movement of our legs would not matter to a fourth order interaction. An interaction device that measures this forward displacement of our body through physical linkages or magnetic trackers would be a fourth order interaction device. The movement of the linkages of

such a device would not be caused directly by the movement of our limbs but rather indirectly. Figure 10 shows how the extension of the right leg acts against the ground, resulting in a ground reaction force that propels the body upwards. If an interaction device were designed to sense the upward momentum of the body using inertial sensors at the hips, it would qualify as a fourth order interaction device. On the other hand, if the device measured the pressure of the right foot against the force platform, the device would qualify as a third order interaction device.

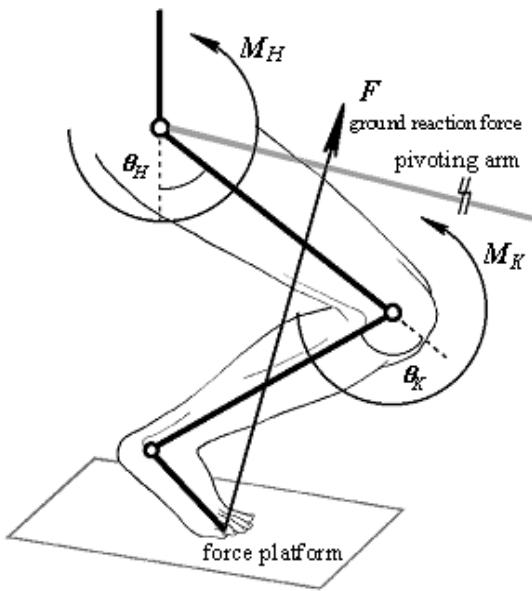


Figure 10. Physical interaction between right leg and ground

The characteristic of the human body a fourth order interaction device senses does not necessarily have to be indirectly caused by the independently articulated limbs. For example, the transfer of energy from our vocal cords into sound waves would qualify as a fourth order of immediacy, thus making cell phones which recognize voice commands fourth order interaction devices.

## **6. Implications**

With the revelation that interaction begins with an electrochemical signal in the brain instead of simply at the interaction device, we already see structure begin to form in the classification of interaction devices. For example, fourth order interaction devices using magnetic trackers, due to their inherent design limitations, must overcome anomalies associated with distance before they can perform similarly to devices using physical contact with the body as their cue. For example, with Polhemus Fastrak magnetic trackers, distortions in the magnetic field of the environment caused by electrical devices results in sensor drift and, thus, inaccurate readings [10]. These errors do not exist in third order interaction devices using mechanical linkages in physical contact with the body, however, the mechanical linkages themselves introduce their own set of difficulties due to mechanical complexity and associated weight. Finally, second order interaction devices using sensors maintaining constant contact with the body avoid the problems associated with third and fourth order interaction devices altogether, but then must rely on complicated algorithms to make educated guesses as to the intentions of the user. Thus, just by the categorization of an interaction device by order of immediacy, we begin to see the limitations a device must either overcome or accept if used in a VE.

## **D. DEGREES OF INPUT, [DI]**

At each of the orders of immediacy, various path types and paths exist to convey information to the interaction device. The various paths used to convey information to the interaction device each constitute a degree of input. The domain of inputs available depends upon what order of immediacy the device uses. The following sections discuss the degrees of input available at each order of immediacy.

## **1. Physiological Inputs**

The original signal within the brain, while initially electrical, triggers chemical responses in the form of neurotransmitters at the synapses. Chemical signals are much less responsive than electrical, but just as significant. Adrenaline, for example, has the ability to increase heart rate, temperature, induce sweating, etc. Thus, at the first order of immediacy corresponding with the physiological level, we see at least two path types within that level, electrical and chemical. In addition, there may exist numerous actual paths for each path type. For example, the adrenal gland releases adrenalin throughout the body to increase both aerobic and anaerobic capacity while, at the same time, the muscles themselves secrete lactic acid in response to muscle breakdown. If we had sensors that could detect the presence of those chemicals in the body, we would be able to use their presence as inputs. Thus, adrenaline and lactic acid each constitute a distinct path of the chemical path type, within the physiological order of immediacy, and are each considered a possible degree of input for use by an interaction device. For the purposes of this work, I do not attempt to enumerate the degrees of freedom available within the first order of immediacy.

## **2. Physical Inputs**

The physical manipulation of the body's skeletal structure provides physical input for both second and third order interaction devices as well as fourth order interaction devices where physical movement is required. This is so because the physical manipulation of the body's skeletal structure is the furthest signal from an original thought signal that the human body can directly provide for use in an interaction device. However, assuming every single muscle or skeleton in the human body as a possible input for interaction devices would be impractical due to the sheer number of muscles and bones in the human musculoskeletal system. More importantly, however, not every muscle or skeleton in the human musculoskeletal system is articulated well enough for the purposes of input. For example, when we look at the bones of the hands, we see

that each hand has 27 bones: the carpus, or wrist, accounts for 8; the metacarpus or palm contains 5; the remaining 14 are digital bones (fingers and thumbs). The 14 digital bones, also called phalanges, or phalanx bones, consist of 2 in each thumb (the thumb has no middle phalanx) and 3 in each of the four fingers. The three bones in each of the four fingers are the distal phalanx, the middle phalanx and the proximal phalanx. Figure 11 shows the various bones in the hand.

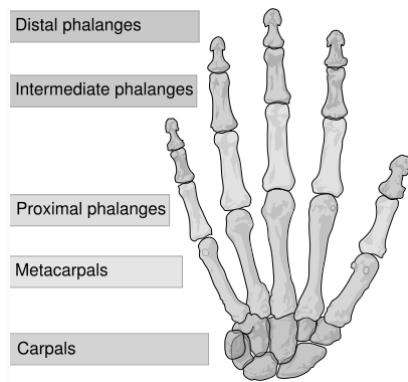


Figure 11. Bones of the hand

The articulation of the human hand is more complex and delicate than that of comparable organs in any other animals. Despite the amazing complexity available in the hands, not every bone is created equally. Each of the bones varies in levels of articulation. The levels of articulation, from greatest to least, are as follows:

- metacarpophalangeal articulations
- interphalangeal articulations of hand
- intercarpal articulations

In the human hand, the metacarpophalangeal joints provide the greatest degree of articulation including flexion, extension, adduction, abduction, and circumduction. The interphalangeal joints only provide flexion and extension. The intercarpal articulations provide the least amount of articulation. The difference between fully articulated joints and those with less articulation, is that

the fully articulated joints provide the greatest level of interaction and are the greatest determinants in physical interaction. The non-fully articulated joints and limb segments after a fully articulated joint are used merely as extensions of the attached limb segment. For this reason, movement of the fully articulated limb segments are what we must consider for practical use in interaction devices, and thus, the assumptive model for the degrees of input considered in this work within the second, third, and fourth orders of immediacy requiring physical movement are the fully articulated limb segments of the musculoskeletal system. The fully articulated limb segments from the center of the body, or the waist, are the head, the legs, the feet, the arms, the hands and the fingers. Figure 12 shows the 19 fully articulated limb segments of the human body.

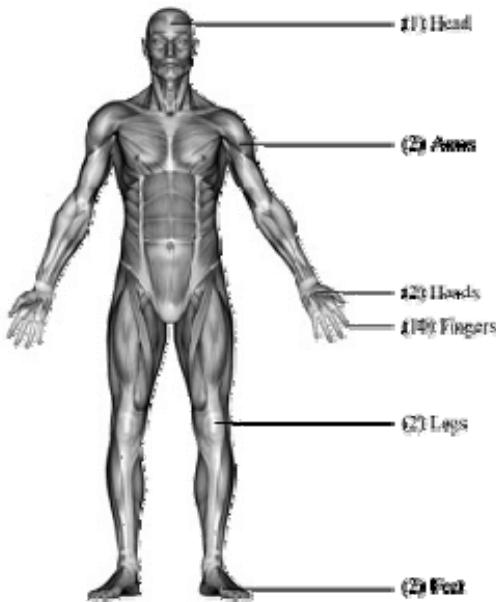


Figure 12. Fully articulated limb segments of the body

### 3. Extra-Physical Inputs

For the fourth order of immediacy, interaction devices do not rely on physical contact with the limbs of the body, however, the manipulation of the limbs of the body may indirectly influence what a fourth order interaction device senses. As discussed previously, fourth order interaction devices requiring

movement of the limbs of the body also use the 19 degrees of freedom derived from the fully articulated limb segments of the human body as their input domain. However, a fourth order interaction device may also sense commands from a user's voice or other characteristics the human body as a whole provides, such as weight. For the purposes of this work, I do not attempt to enumerate the degrees of freedom available within the fourth order of immediacy except to say that they neither rely on a physiological signal nor are directly caused by the manipulation of the independently articulated limbs.

## **E. DEGREES OF OUTPUT, [DO]**

The degrees of output correlate to the possible functions the interaction results in. For example, in a 35mm single lens reflex (SLR) camera, the traditional functions would consist of adjusting lens aperture, shutter speed, and focus, and possibly exposure meter adjustments. In good camera design, all of these functions can be manipulated quickly, without forcing a photographer to remove their gaze from what they are photographing or change their grip. Typically, a photographer will be able to manipulate aperture using their right index finger on a dial at the front of the camera, while manipulating shutter speed using their right thumb on a dial at the back of the camera. Finally, focus is typically manipulated by rotating the lens focus ring between the left index finger and left thumb while the left hand provides a stable platform beneath the camera. We see in the design of this simple SLR camera, a 1:1 mapping of aperture and shutter speed manipulations to the index finger and thumb, and a 2:1 mapping of focus manipulations. We see that focusing takes up a greater degree of input, limiting additional simultaneous manipulations. Thus, the mapping of each degree of input to its corresponding degree of output provides, in itself, insight into the usability of an interaction device.

For locomotion, the types of movements required for military operations on urbanized terrain (MOUT), represents the most demanding set of functions we can expect to encounter for use in virtual environments. Previous work by

Unguder [11] in his task analysis of building clearance operations, determined the following movements as requirements [11]:

- Walking
- Sidestepping
- Kneeling
- Crawling
- Jumping
- Rolling
- Running
- Looking around the corner
- Backward movement
- Lie prone

Using this analysis, we categorize each of these movements in terms of velocity with respect to viewpoint and make each of the categorizations a degree of locomotion output. Walking, running, kneeling, and crawling are merely translational movements along a surface at different viewpoint levels and speeds. For each distinct speed and viewpoint level, we assign four degrees of freedom to account for forward, side, and backward movements for a total of 16 degrees of freedom. We also consider jumping to have the same four degrees of freedom, plus an additional degree when jumping with no lateral displacement, for a total of 5 degrees of freedom. Leaning left and right (or Ungader's "looking around a corner") provides an additional 2 degrees of freedom. Finally, rolling left and right provides the final 2 degrees of freedom. Altogether, this categorization of the movements required in building clearance adds up to 25 degrees of freedom, or output, and we call this assumptive model the *25 Degree Model of Locomotion*. Coincidentally, these are the standard movements used by popular FPS video games.

- 4 DOF for running
- 4 DOF for walking (forward, sidestep left and right, and backstep)
- 4 DOF for kneeling
- 4 DOF for crawling
- 5 DOF for jumping
- 2 DOF for leaning
- 2 DOF for rolling when prone only

## F. OVERALL TAXONOMY

Figure 13, below, shows the complete structure of the taxonomic structure. As we see, classification is based on the mapping of input to output on a per-signal-path basis. What this taxonomy facilitates is the ability to compare the effectiveness of different signal paths for the same output on a limb-by-limb basis in the case of 2<sup>nd</sup> and 3<sup>rd</sup> order devices. By using Fitts' law and other human performance models or measures, we may then be able to isolate interactions into basic units of psychomotor action and have a good idea as to a device's usability for a particular interaction merely by its mapping within the taxonomic structure.

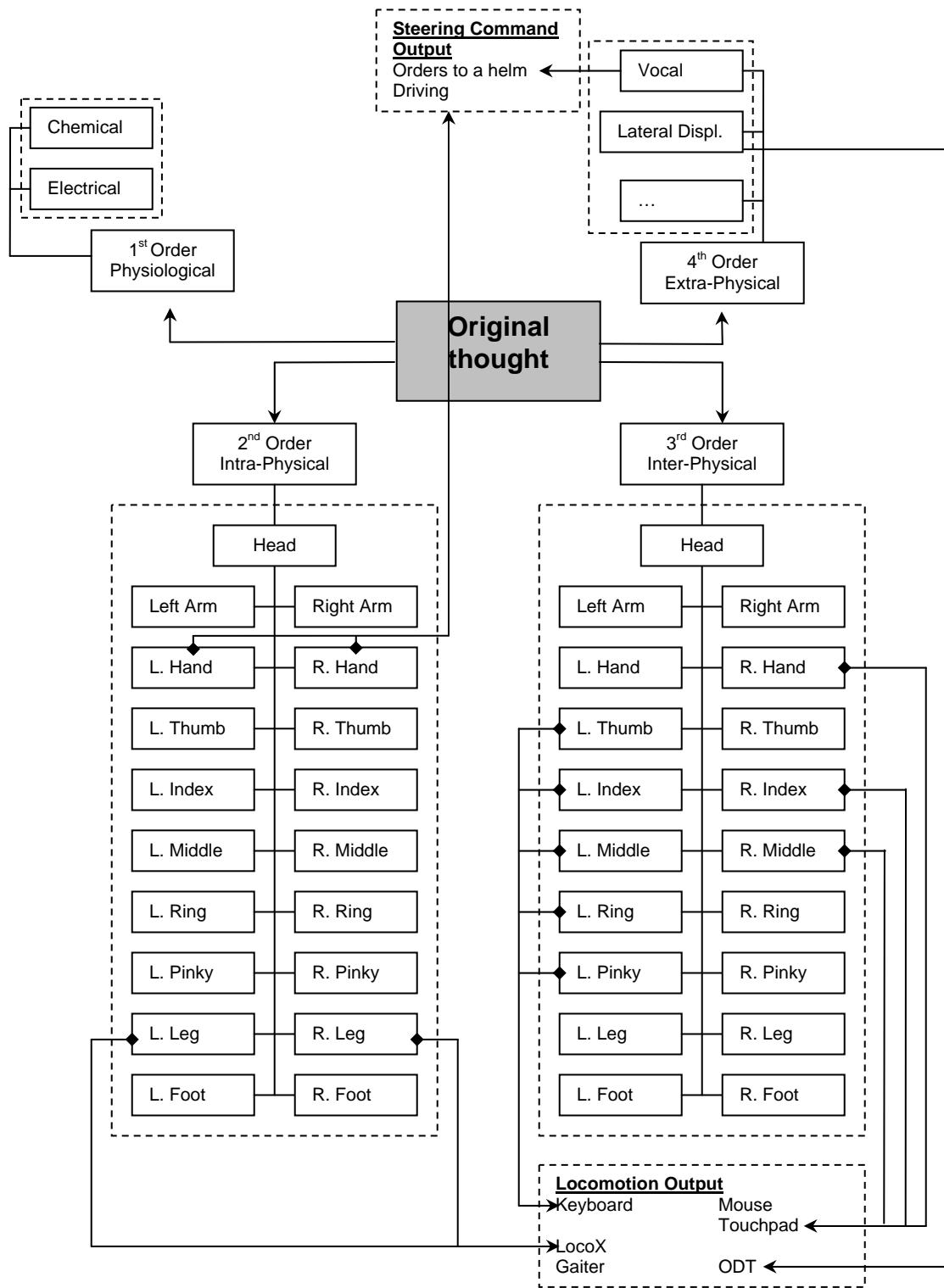


Figure 13. Proposed taxonomy of human interaction

## 1. Metrics

The following quantitative measures of an interaction device are based solely on the device's location in the taxonomy. They are provided as possible candidates for measures of performance.

### a. *Order of Immediacy, [OI]*

The order of immediacy, itself, may be a possible indicator of usability. It may even be possible to break each order of immediacy into finer divisions based on actual numbers of muscles used to cause an interaction, for example, the coordination required in the supplementary motor area of the cortex which coordinates movements between two separate limbs, or the coordination between segments of the same limb. Similar to the Accot-Zhai steering law's integral of Fitts' law movements, increased coordination of muscle movements may prove more difficult to perform, whereas simple, flexion and extension movements, such as those of the upper arms, may prove simpler to perform.

### b. *Ratio of Output to Input, [DI:DO]*

The ratio DO:DI essentially shows the tradeoff of efficiency to over-tasking. If an interaction uses too many degrees of input for a single function, it requires a greater degree of muscle coordination and may be more difficult to use. On the other hand, if too many functions are assigned to a single degree of input, the time it takes to decide which function of the many to perform increases logarithmically, according to Hick's law, shown in Figure 14.

$$T = b \log_2(n + 1)$$

Figure 14. Hick's law

Interestingly, Hick's law bears striking resemblance to Fitts' law and serves as yet another example of the connection between cognition and the human motor system, which this human-centric taxonomy attempts to take advantage of.

**c. Degrees of Input Conserved, [DI<sup>C</sup>]**

DI<sup>C</sup> is related to DO:DI in that an interaction device with a high ratio leaves less degrees of input available for additional simultaneous interactions. For example, in the example of the SLR camera, mentioned earlier, because both the index finger and thumb of the left hand are used for focusing, the ability to simultaneously adjust some other setting, such as sensitivity levels, would require a modality shift. For this reason, cameras with autofocus free up the left hand entirely, and thus, have a greater DI<sup>C</sup> than manual focus cameras. The same goes for locomotion devices, where joysticks and control pads used by the hands to control locomotion then decrease the ability to use the hands for things that would naturally be performed by them.

**G. NAMING INTERFACES USING PROPOSED TAXONOMY**

**1. Suggested Naming Convention**

The suggested naming convention using this taxonomy would be to use every classification level of the interaction device in order as shown below:

“[OI]<sup>st/nd/rd/th</sup> order,

[DI] degree (noun name of input type, if available) input,

[DO] degree (noun name of output type, if available) output interface.”

**2. Suggested Abbreviation Rules**

The following rules may be used to abbreviate naming when parts of the name are implicitly suggested in adjacent identifiers within the name:

**Rule 1.** For the degrees of input [DI] and output [DO], you may omit the word “degrees” if there exists a noun name for the input or output type

**Rule 2.** If there is only 1 degree of input, you may omit the number of degrees [DI] or [DO] altogether, as long as the input or output type implicitly infers only 1 degree of input or output. For example, the human body only has 1 head.

**Rule 3.** You may omit the word “output” if the noun name itself is sufficient.

### 3. Examples

I list several examples below of common interaction devices using the proposed naming convention:

**Example 1.** Treadmill platform using displacement from center as cue for speed: “4<sup>th</sup> order, 2 leg input, forward only locomotion interface”

**Example 2.** Omni-Directional Treadmill: “4<sup>rd</sup> order, 2 leg input, full walking locomotion interface”

**Example 3.** Game console pressure-pad for use in “Dance, Dance Revolution” video game: “3<sup>rd</sup> order, 2 leg input, left-right output interface”

THIS PAGE INTENTIONALLY LEFT BLANK

## **IV. IMPLEMENTATION**

### **A. DETERMINATION OF A CUE**

Based on the structure of the taxonomy proposed in the last chapter, as well as extensive research on human, rapid locomotion, it was determined that the 2nd order cues provided by the upper legs, by themselves, are able to support recognition of 23 of the 25 locomotion functions (all but leaning) using only 2 sensors. Quantitatively, we determine a device relying on such an input to have a DO:DI ratio of 11.5, which compares favorably to the mouse and keyboard implementation, which is calculated to have a ratio of 2.78 by dividing 25 degrees of output (includes leaning) using a total of 9 degrees of input, which includes all of the fingers of the left hand to press the assigned keys for the various speeds and viewpoint levels, and 3 of the fingers of the right hand as well as the hand itself to hold the mouse and change the direction of the movement.

### **B. 2<sup>ND</sup> ORDER, 2 LEG LOCOMOTION RECOGNITION ALGORITHM**

With the 2<sup>nd</sup> order cue determined in the last section, which provides information in the form of Euler angles to two inertial sensors attached to the upper legs, we create an algorithm which recognizes walk in place movements as metaphors for locomotion in a virtual environment. We call this algorithm designed for a 2<sup>nd</sup> order, 2 leg input locomotion device 22-LRA for brevity purposes

#### **1. General Design**

22-LRA can be thought of as a “black box” with Euler angles as inputs and movement instructions as output. Sensors attached to a user’s upper legs, as shown in Figure 15, provide the Euler angles for the algorithm, which analyzes the information and determines the position and velocity of the legs. From this

determination, a movement such as walk, sidestep left or right, or back step instructs the user application how to control the movement of the user viewpoint.

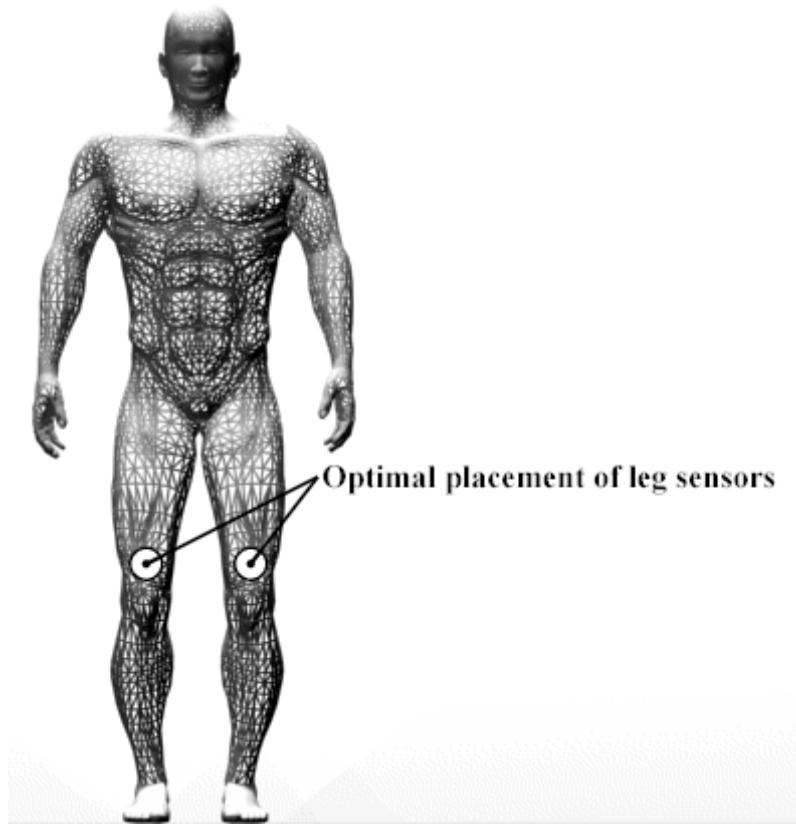


Figure 15. Placement of leg sensors

The simple requirements of 22-LRA (yaw, pitch and roll from two sensors) along with the direct coupling of the sensors to the fully articulated limbs, a characteristic of all 2<sup>nd</sup> order interaction devices, eliminates the need for bulky contraptions and mechanical linkages. 22-LRA also facilitates the use of a wide range of sensors, wired or wireless, with minimal modification to the algorithm's implementation. As long as the sensor outputs yaw, pitch and roll, the algorithm can determine locomotion. To illustrate this, the algorithm was implemented almost entirely before any sensors had even been procured for testing. When actual sensors were finally obtained, it was an insignificant task to integrate those particular sensors with the implemented code using the original equipment manufacturer's device calls.

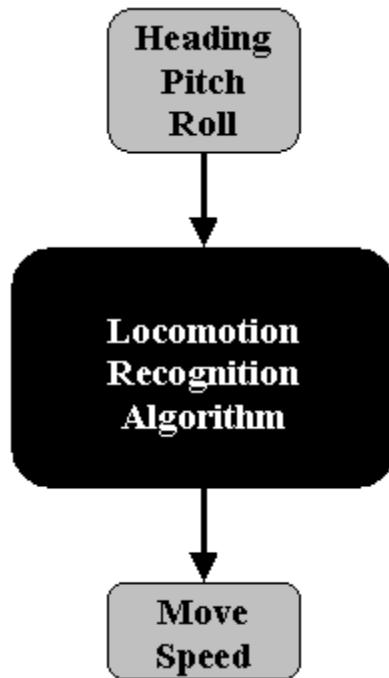


Figure 16. 22-LRA model

## 2. Internal Structure

Internally, 22-LRA uses a simple structure with a complicated “payload”. This simple structure can best be described using a state-transition diagram. State-transition diagrams, used in unified modeling language (UML) are extremely useful tools in organizing the relationships between the states in a system. In a state-transition diagram, systems are represented by all the states an object can have, the events under which an object changes state (transitions), the conditions that must be fulfilled before the transition will occur (conditions), and the activities undertaken during the life of an object (actions). Figure 17 shows a state-transition diagram representing an auction.

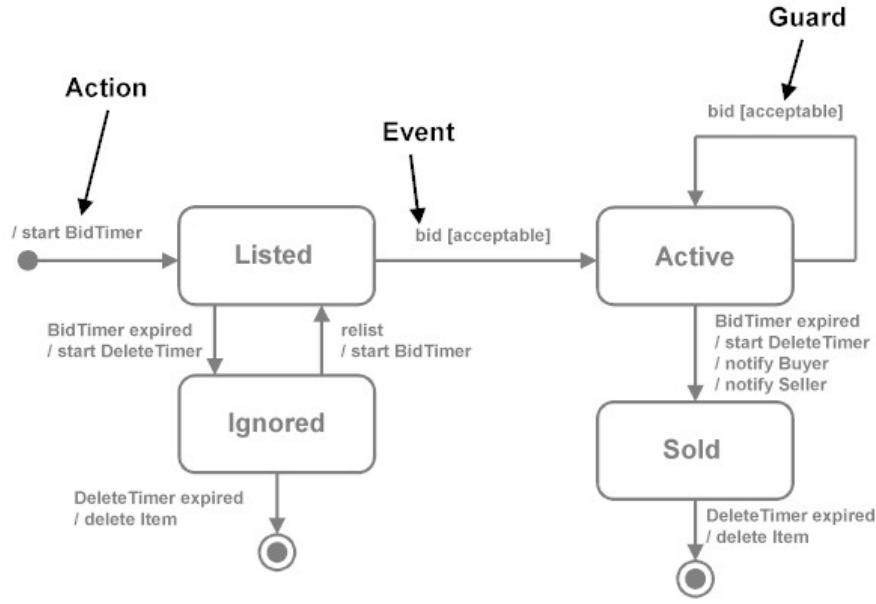


Figure 17. Example of state-transition diagram for auction process

Like the above example, 22-LRA can be represented as a group of states linked together by transitions and actions. Figure 18, below, shows the various states traversed by the algorithm.

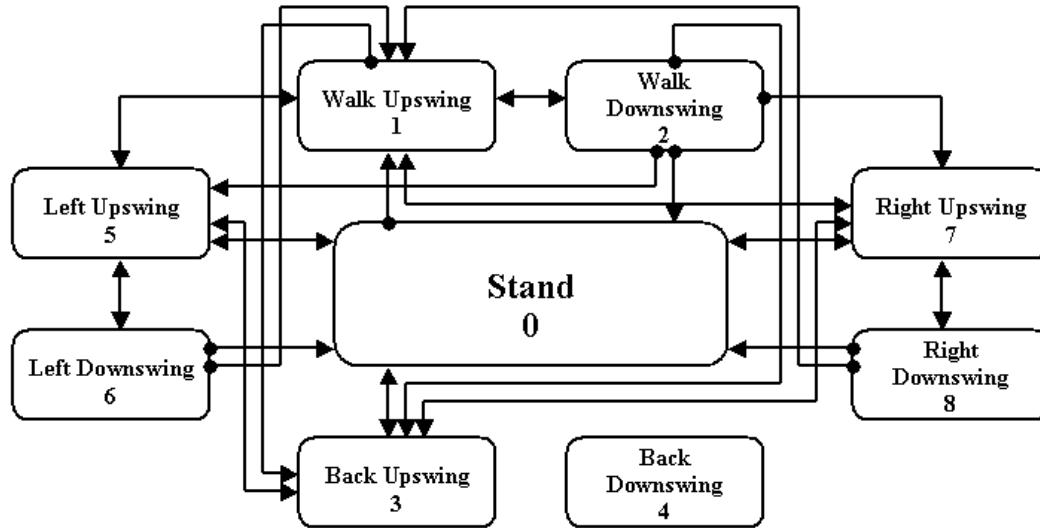


Figure 18. Overall state-transition diagram

As you can see from Figure 18, the algorithm essentially divides lateral motion (forward, left, right and back) into upswing and downswing phases, each

of these constituting a single state along with the neutral state of stand, for a total of 9 discrete states. The algorithm does not distinguish rotational motion such as turning. Rotational movement is determined independently from the various translational movements 22-LRA recognizes. Ideally, rotation would be implemented using a third sensor, such as that on a tracked, head-mounted display, or by using the average of the yaw data between both of the existing leg sensors, to facilitate the decoupling of movement direction from viewpoint direction.

### **3. The Payload**

The numerous arrows linking each of the states in Figure 18 represent the complicated “payload” mentioned earlier. The “payload” essentially encodes the results of lengthy background research on human locomotion, in particular that of the gait and changes in gait as the body changes modes of locomotion. The payload also encodes constraints on transitions that are impossible or highly unlikely in human locomotion based on the understanding of locomotion as a controlled fall in the intended direction of travel. Each of the transitions represents the satisfaction of one of almost 2,000 conditional statements regarding leg positions and velocities. A 3-dimensional matrix described by the current state, leg position, and leg velocity best represents the “payload”. Appendix A shows, in greater detail, how each arrow in Figure 18 is actually the satisfaction of one of several conditions for transition.

## **C. ALGORITHM STEPS**

Now that we have a good understanding of the general design and structure used in 22-LRA, we will examine its actual steps. The algorithm consists of five basic phases. The phases are as follows:

- Analyze current leg positions
- Analyze current leg velocities
- Identify possible new movement type

- Set new movement type
- Set new movement speed

We now describe each phase of the algorithm, beginning with the analysis of leg positions.

## 1. Analyze Leg Positions

The steps in the first phase of 22-LRA involve identifying the positions of each of the legs and then identifying which of the position situations that the combination of both leg positions satisfies.

### a. *Identify Left Leg Position*

The left leg position is determined by analyzing the pitch and roll information of the left leg sensor. To identify the position of the leg, we construct a graph with roll describing the x-value and pitch describing the y-value, shown in Figure 19. We divide the graph into sectors describing when the leg is forward, backward, left, or right. We also leave an area in the center for when the leg is straight, the width of which serves as a threshold. You may notice when the left leg is in the right sector, it is assumed to be straight. We eliminate the possibility of the left leg being in the right sector because, unless the legs are crossed, it would create a position of imbalance that is not typically used for locomotion.

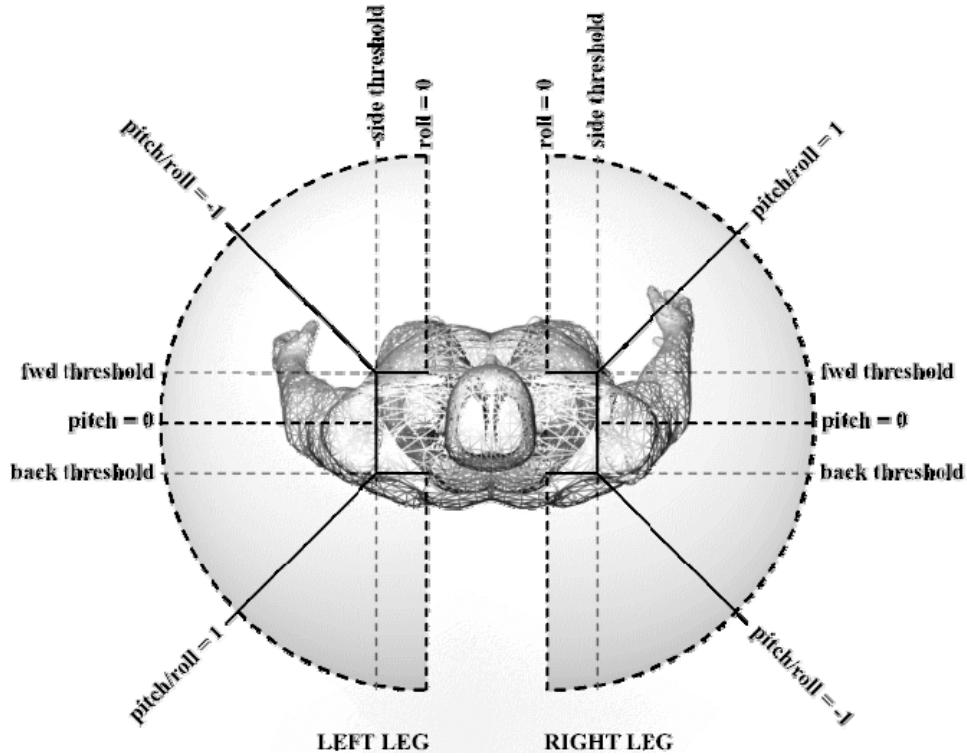


Figure 19. Graph describing how the position of each leg is determined by roll and pitch.

It is worth mentioning that we may also be able to use heading information to improve our analysis of the leg positions. To illustrate why, assume a user were to point their left foot to the left and then step outward, which is a normal motion involved in human gait. Instead of the leg position being sensed as left, it would be sensed as being forward due to the change in the heading and the position of the sensor on the user's thigh. Using heading information could improve recognition of leg positions by recognizing this.

#### **b. Identify Right Leg Position**

The right leg position is determined the same way as the left leg, the only difference being that the right leg is assumed straight when it is in the left sector.

**c. Identify Leg Position Situation Satisfied by the Combination of Left and Right Leg Positions**

Now that we have identified the individual positions of each leg, the final step in the first phase of the algorithm is to identify which of 25 position situations the combination of leg positions satisfies. This can be done by first creating a matrix with the possible left leg positions along the vertical axis and the possible right leg positions along the horizontal axis, as in Table 1. For each leg, 0 represents a straight leg, 1 represents the leg lifted forward, 2 represents the leg extended backward, 3 represents the leg extended outward, and 4 represents the leg extended inward.

R	0	1	2	3	4
L	○	↑	↓	→	←
0	○	0	4	5	6
1	↑	1	7	8	-
2	↓	2	9	10	-
3	←	3	-	-	11
4	→	-	-	-	-

Table 1. Position situation matrix

In the matrix, you may notice that 13 of the possible combinations are not used. Certain combinations cause imbalanced positions that a user would not typically perform if trying to walk in place. The imbalanced positions are reached when, the legs are forced to cross one another, which only happens if one of the legs are extended inward, or the positions of the legs are not along the same plane bisecting the center of the body, for example, if the left leg is forward and the right leg is to the right. This combination of leg positions would cause the user to fall back and to the left. In general, the legs act as scissors so that at any one time, the user's torso and each of the legs lie on a single plane. The planes of the body are shown in Figure 20. If a position of imbalance is detected, however, we must still treat that situation, and we do so by assuming both legs are in the neutral position, 0, despite the matrix representing those situations with a dash.

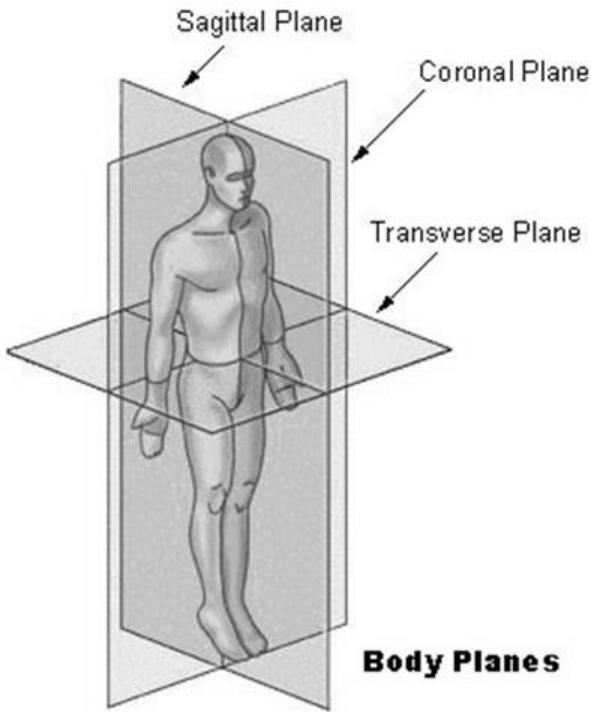


Figure 20. Illustration of frontal (coronal) and sagittal planes

## 2. Analyze Leg Velocities

### a. *Identify Velocity of Left Leg*

The first step in the second phase of 22-LRA is to determine the velocity of the left leg. The velocity is simply obtained by finding the last position of the leg and the current position, and then dividing the difference by the time that has passed since the last information was obtained. The velocity is obtained separately for pitch and roll. As with the analysis of leg positions, heading information is not needed since only pitch and roll are used in the algorithm. There are two types of velocity. Pitch velocity correlates to frontal velocity and roll velocity correlates to sagittal velocity. Only one of these velocities is considered at any one time. The type of velocity that is considered depends on the position of the leg. If the combination of both legs is parallel to the frontal plane, i.e., in sector 1 or 2 of Figure 20 (plane extending forward and backward that bisects the left and right halves of a human), the type of velocity that is

considered is frontal, i.e., only pitch velocity is considered. If the combination of both legs is in one of the sagittal sectors, then the type of velocity that is considered is sagittal, i.e., only roll velocity is considered. If both legs are straight, in sector 0 of figure 3, the type of velocity that is the greatest is the one that is considered. Figure 21, below, represents the possible velocity situations.

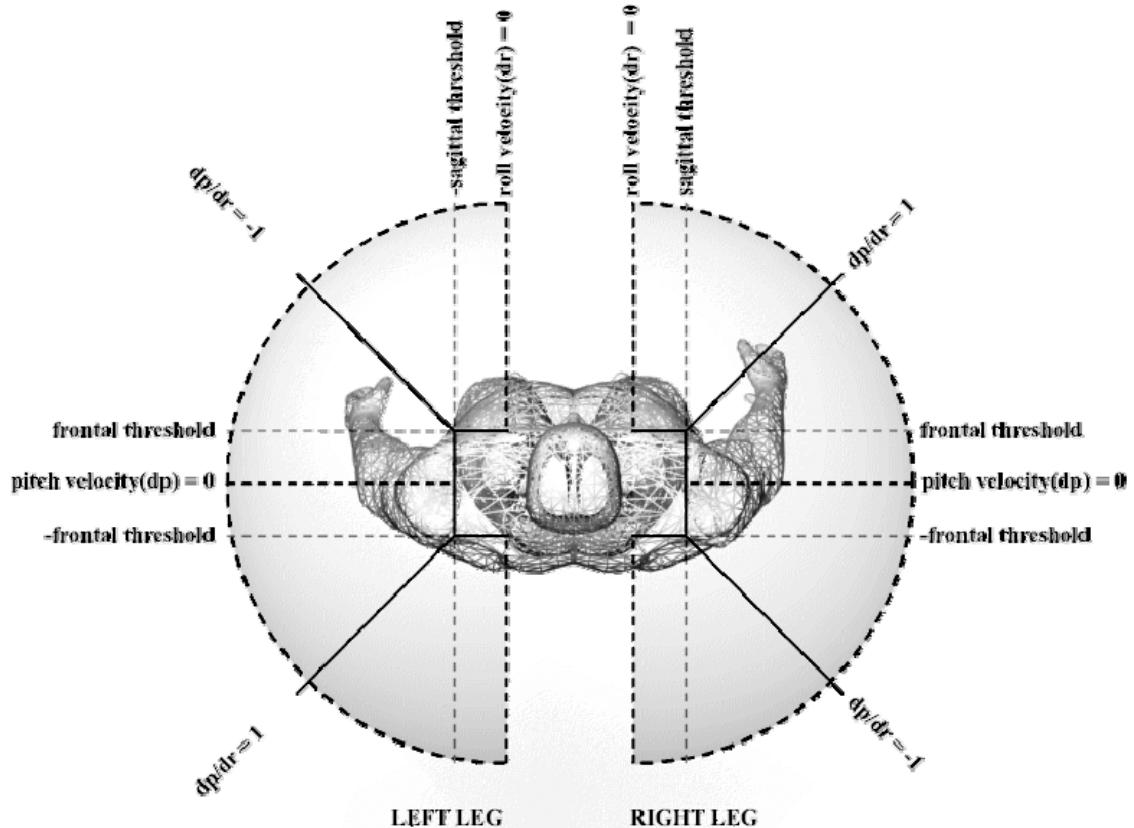


Figure 21. Graph describing the relationship between pitch and roll velocity

As you can see, the graph representing the velocity sectors is similar to the one representing position sectors. The main difference is that, now, we consider all sectors for each leg instead of ignoring the inward sector since inward velocities do not necessarily create situations of imbalance.

### **b. Identify Velocity of Right Leg**

The velocity of the right leg is found the same way as the left leg.

**c. Identify Leg Velocity Situation (VS) from One of 17 Situations using Velocity Matrix (VM)**

Based on the combination of leg velocities, as shown in Table 2, 22-LRA determines one of 17 possible velocity situations. As with leg positions, leg velocities can only be frontal or sagittal, not a combination of both. Combinations not falling into these categories must be resolved by ignoring one of the legs, with the leg having frontal velocity taking priority.

R	0	1	2	3	4
L	○	↑	↓	→	←
0	○	5	6	7	8
1	↑	1	9	10	-
2	↓	2	11	12	-
3	←	3	-	-	13 14
4	→	4	-	-	15 16

Table 2. Velocity situation matrix

**3. Identify New Movement Type**

Now that we have determined both the position situation and velocity situation, we determine whether those situations satisfy a condition to transition to a new movement state from the previous movement state. The transition matrix, shown in Table 3, shows the possible changes based on the linearized combinations of leg positions (rows) and velocities (columns) when we are in the stand state. The blacked out cells of the matrix represent combinations that are either impossible or highly unlikely due to the nature of our bodies, thus resulting in no change, whereas the stippled cells represent no change due to an intelligent assumption based on the current movement. Appendix B contains notes on observations during research of rapid human locomotion that led to those assumptions.

Move: Stand

P	V	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	○○	0	0	0	0		0	0	0		0	0	0	0				
1	↑○		1								1	1						
2	↓○			3							3	3						
3	←○				5				5					5	7	5		
4	○↑					1					1	1						
5	○↓						3				3	3						
6	○→				7			7						7	7	5		
7	↑↑																	
8	↑↓										3							
9	↓↑										3							
10	↓↓																	
11	↔				5			7										

Table 3. Transition matrix for stand state

#### 4. Check and Set New Movement State

Once the new movement state is determined, it is set as the current state.

#### 5. Use Conditionals to Determine and Set New Movement Speed

We then must determine the speed of the movement. We can use the velocity of the movement of the legs, the maximum height of the legs, or a combination of both. 22-LRA uses a combination of both to determine movement speed. Either way, more effort must be expended to increase speed.

### D. ILLUSTRATION OF ALGORITHM USAGE

To illustrate the algorithm in action, we will use the example of a person making one full step in place. We divide the user's steps into two phases, the upswing phase and the downswing phase. The beginning state of the user is that he is standing, which is state 0.

#### 1. Upswing Phase

The user begins to lift his left leg while the right leg remains straight. The algorithm determines the positions of the legs with the left leg satisfying position 1 and the right leg satisfying position 2. As shown in Table 1, the algorithm determines the combination of leg positions to result in position situation 1.

After determining the position situation, the algorithm then determines that the left leg velocity is moving forward, and thus, satisfies situation 1, while the right leg is stationary, thus satisfying situation 0. As shown in Table 2, the algorithm determines the combination of leg velocities to result in velocity situation 1.

The algorithm then combines the results of position and velocity for both legs and using the transition matrix for the current state, *stand*, determines which state to transition to. In this case, Table 3 shows the algorithm determines the user transitioned from state 0 to state 1, which is the forward upswing phase.

The algorithm then determines and sets the speed. For initial movements, the algorithm begins with the walking speed, but increases after capturing the peak height of the lifted leg, which does not occur until the downswing phase. In this algorithm, the peak height used to trigger increased speed can be adjusted to match the gait of a particular user when calibrating the device prior to initial use, however, that feature has not yet been implemented.

		Move: Fwd Upswing																
P	V	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	OO	0	0	0	0		0	0	0		0	0	0	0				
1	↑O	1	1	2			1	1			1	1	2	2				
2	↓O			3								3	3					
3	←O				5				5						5	7	5	
4	O↑	1	1	1			1	2			1	2	1	2				
5	O↓						3				3	3	3					
6	O→				7			7							7	7	5	
7	↑↑																	
8	↑↓										3							
9	↓↑										3							
10	↓↓										3							
11	↔				5			7										

		Move: Fwd Downswing																
P	V	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	OO	0	0	0	0		0	0	0		0	0	0	0				
1	↑O	2	1	2			2	2			1	1	2	2				
2	↓O			3								3	3					
3	←O				5				5						5	7	5	
4	O↑	2	2	2			1	2			1	2	1	2				
5	O↓						3				3	3	3					
6	O→				7			7							7	7	5	
7	↑↑																	
8	↑↓										3							
9	↓↑										3							
10	↓↓										3							
11	↔				5			7										

Table 4. Transition matrix for forward upswing and downswing states

## **2. Downswing Phase**

The user now begins to lower their leg. The position has not yet changed, and thus remains in position situation 1. However, the leg velocity changes such that the left leg now has a backward velocity while the right leg remains stationary. The algorithm determines this combination to satisfy velocity situation 2. The transition matrix shown in Table 4 determines the combination of position situation and velocity situation to result in a transition from state 1 to state 2, which is the forward downswing phase.

An important thing to note about downswing phases is that transitions to other movements cannot occur until a user is in a downswing phase. This and other constraints on transitions keep the algorithm from somehow falsely interpreting sensor errors as locomotion. See Appendix B for notes on these constraints.

After setting the downswing phase, the algorithm senses the peak height of the leg lift and determines whether it was greater than the height set for the running state. If so, it then increases the speed of the forward movement.

Finally, when the left leg momentarily becomes stationary and the right leg then commences lifting to transition into the forward upswing phase, the algorithm uses an adjustable delay to prevent transition to a stand state, allowing continuous movement despite constant transitions between upswing and downswing phases.

## **E. CONCLUSION**

By beginning the development of a locomotion device with a 2<sup>nd</sup> order cue using the proposed taxonomy in the previous chapter, we were able to determine a cue that required less mechanical linkages and associated weight than other locomotion devices. By seeing the wealth of information the combination of leg positions and leg velocities can provide to a locomotion device, we were then able to create an algorithm which can distinguish between 23 of the 25 possible

locomotion functions required for complicated tasks such as building clearance operations which soldiers are regularly conducting in Iraq and Afghanistan. Finally, through intensive research and understanding of natural locomotion, we were able to constrain transitions in order to avoid false cues from sensor errors, thereby creating a robust locomotion device.

THIS PAGE INTENTIONALLY LEFT BLANK

## V. PRELIMINARY RESULTS

In order to validate the effectiveness of the 2<sup>nd</sup> order, 2-leg locomotion device, which we shall call LocoX for brevity purposes, we compare it to the Omni-Directional Treadmill [13] by conducting a Human Ability Requirements (HAR) analysis using Fleischman's Job Analysis Survey (F-JAS). The results of F-JAS agree with the analysis of LocoX in the context of the proposed taxonomy of this work. By looking at a device in the context of the human abilities it requires in its use, we may quantitatively compare interaction devices by their similar components. Following the HAR analysis, we then performed initial tests with 2 subjects, 1 male and 1 female. Figure 22 shows the implementation using 2 Intersense inertial sensors on the legs and 1 on a head-mounted display. The video was further projected onto a screen for viewers to see what the user saw.



Figure 22. Illustration of implemented demo in use

## A. HUMAN ABILITY REQUIREMENTS ANALYSIS USING F-JAS

In Tables 5 through 8, we took the results of a sample HAR analysis done by Cockayne and Darken [13] comparing the ODT to natural human locomotion. This analysis was done by using F-JAS as it applied to natural locomotion and locomotion in a VE using the ODT. After reviewing the scaling used in this analysis, we performed an analysis of LocoX. Using a thorough study of anatomy, both dynamic and static, we then compared LocoX to natural locomotion. The results of this analysis are shown in Figures 9 through 10. A brief description of the considerations taken into account for each score in the analysis is given, followed by a conclusion of the comparison to the ODT.

	1. Multilimb Coordination	2. Rate Control	3. Response Orientation	4. Reaction Time	5. Speed of Limb Movement	6. Static Strength	7. Explosive Strength	8. Dynamic Strength	9. Trunk Strength	10. Extent Flexibility	11. Dynamic Flexibility	12. Gross Body Coordination	12(A). Side-to-Side Coordination	12(B). Front-to-Back Coordination	12(C). Rotational Coordination	13. Gross Body Equilibrium	13(A). Side-to-Side Equilibrium	13(B). Front-to-Back Equilibrium	13(C). Rotational Equilibrium	14. Stamina	15. Speed of Transport
Accelerate from Rest to Walk or Jog	●			●	●	●	●	●	●	●	●	●	■	■	■	●	■	■	■	●	
Decelerate from Walk or Jog to Rest	●			●	●	●	●		●	■	■	■	■	■	■	●	■	■	■	●	
Accelerate from Walk to Jog	●			●		●	●	●	●	■	■	■	■	■	■	●	■	■	■	●	
Decelerate to Walk from Jog	●			●		●	●	●	●	■	■	■	■	■	■	●	■	■	■	●	
Walk	●			●		●	●	●	●	●	●	●	■	■	■	●	■	■	■	●	
Jog	●			●		●	●	●	●	●	●	●	■	■	■	●	■	■	■	●	
Turn in Place (no forward or side movement)	●				●		●	●	●	●	■	■	■	■	●	■	■	■	■		
Side Step	●				●		●	●	●	●	■	■	■	■	●	■	■	■	■		
Tilt Upper-Body (without Foot Movement)	●				●		●	●	●	●	■	■	■	■	●	■	■	■	■		

Table 5. Absense/presence analysis of natural locomotion

	1. Multilimb Coordination	2. Rate Control	3. Response Orientation	4. Reaction Time	5. Speed of Limb Movement	6. Static Strength	7. Explosive Strength	8. Dynamic Strength	9. Trunk Strength	10. Extent Flexibility	11. Dynamic Flexibility	12. Gross Body Coordination	12(A). Side-to-Side Coordination	12(B). Front-to-Back Coordination	12(C). Rotational Coordination	13. Gross Body Equilibrium	13(A). Side-to-Side Equilibrium	13(B). Front-to-Back Equilibrium	13(C). Rotational Equilibrium	14. Stamina	15. Speed of Transport
Accelerate from Rest to Walk or Jog	5	1	1	1	5	2	6	6	5	1	4	5	■	■	■	■	■	■	■	2	5
Decelerate from Walk or Jog to Rest	5	1	1	1	5	2	1	5	5	1	4	4	■	■	■	■	■	■	■	1	5
Accelerate from Walk to Jog	5	1	1	1	5	1	2	6	5	1	4	5	■	■	■	■	■	■	■	2	4
Decelerate to Walk from Jog	5	1	1	1	5	1	1	5	5	1	4	4	■	■	■	■	■	■	■	1	4
Walk	5	1	1	1	4	1	1	4	3	1	3	3	■	■	■	■	■	■	■	2	2
Jog	6	1	1	1	5	1	1	4	4	1	4	4	■	■	■	■	■	■	■	4	3
Turn in Place (no forward or side movement)	4	1	1	1	1	2	1	1	3	2	1	2	■	■	■	■	■	■	■	1	1
Side Step	5	1	1	1	1	3	1	1	3	2	1	4	■	■	■	■	■	■	■	1	1
Tilt Upper-Body (without Foot Movement)	2	1	1	1	1	3	1	1	5	1	1	3	■	■	■	■	■	■	■	1	1

Table 6. Scaled analysis of natural locomotion

	1. Multilimb Coordination	2. Rate Control	3. Response Orientation	4. Reaction Time	5. Speed of Limb Movement	6. Static Strength	7. Explosive Strength	8. Dynamic Strength	9. Trunk Strength	10. Extent Flexibility	11. Dynamic Flexibility	12. Gross Body Coordination	12(A). Side-to-Side Coordination	12(B). Front-to-Back Coordination	12(C). Rotational Coordination	13. Gross Body Equilibrium	13(A). Side-to-Side Equilibrium	13(B). Front-to-Back Equilibrium	13(C). Rotational Equilibrium	14. Stamina	15. Speed of Transport
Accelerate from Rest to Walk or Jog	●	●	●	●	●	●	●	●	●	●	●	■	■	■	■	■	■	■	■	●	●
Decelerate from Walk or Jog to Rest	●	●	●	●	●	●	●	●	●	●	●	■	■	●	●	■	■	●	●	●	●
Accelerate from Walk to Jog	●	●	●	●	●	●	●	●	●	●	●	■	■	●	●	■	■	●	●	●	●
Decelerate to Walk from Jog	●	●	●	●	●	●	●	●	●	●	●	■	■	●	●	■	■	●	●	●	●
Walk	●	●	●	●	●	●	●	●	●	●	●	■	■	●	●	■	■	●	●	●	●
Jog	●	●	●	●	●	●	●	●	●	●	●	■	■	●	●	■	■	●	●	●	●
Turn in Place (no forward or side movement)	●	●	●	●	●	●	●	●	●	●	●	■	■	●	●	■	■	●	●	●	●
Side Step	●	●	●	●	●	●	●	●	●	●	●	■	■	●	●	■	■	●	●	●	●
Tilt Upper-Body (without Foot Movement)	●	●	●	●	●	●	●	●	●	●	●	■	■	●	●	■	■	●	●	●	●

Table 7. Absense/presence analysis of VE locomotion on the ODT

	1. Multilimb Coordination	2. Rate Control	3. Response Orientation	4. Reaction Time	5. Speed of Limb Movement	6. Static Strength	7. Explosive Strength	8. Dynamic Strength	9. Truck Strength	10. Extent Flexibility	11. Dynamic Flexibility	12. Gross Body Coordination	12(A). Side-to-Side Coordination	12(B). Front-to-Back Coordination	12(C). Rotational Coordination	13. Gross Body Equilibrium	13(A). Side-to-Side Equilibrium	13(B). Front-to-Back Equilibrium	13(C). Rotational Equilibrium	14. Stamina	15. Speed of Transport
Accelerate from Rest to Walk or Jog	6	4	7	7	5	3	6	7	6	1	4	■	6	6	1	■	5	6	1	4	6
Decelerate from Walk or Jog to Rest	6	4	7	7	5	3	1	6	6	1	5	■	5	7	1	■	4	7	1	1	7
Accelerate from Walk to Jog	6	4	7	7	5	1	3	7	6	1	4	■	6	5	1	■	5	5	1	4	6
Decelerate to Walk from Jog	6	4	7	7	5	1	1	6	6	1	5	■	5	6	1	■	4	6	1	1	6
Walk	6	3	3	4	4	1	1	5	4	1	3	■	5	3	1	■	4	3	1	4	4
Jog	7	2	4	5	5	1	1	5	5	1	3	■	6	3	1	■	4	3	1	6	4
Turn in Place (no forward or side movement)	5	3	3	3	1	3	1	1	4	5	1	■	5	4	6	■	4	4	6	1	1
Side Step	6	3	4	7	1	4	1	1	4	5	1	■	7	6	5	■	7	6	5	1	1
Tilt Upper-Body (without Foot Movement)	3	2	3	6	1	4	1	1	6	1	1	■	4	7	1	■	4	7	1	1	1

Table 8. Scaled analysis of VE locomotion on the ODT

	1. Multilimb Coordination	2. Rate Control	3. Response Orientation	4. Reaction Time	5. Speed of Limb Movement	6. Static Strength	7. Explosive Strength	8. Dynamic Strength	9. Truck Strength	10. Extent Flexibility	11. Dynamic Flexibility	12. Gross Body Coordination	12(A). Side-to-Side Coordination	12(B). Front-to-Back Coordination	12(C). Rotational Coordination	13. Gross Body Equilibrium	13(A). Side-to-Side Equilibrium	13(B). Front-to-Back Equilibrium	13(C). Rotational Equilibrium	14. Stamina	15. Speed of Transport
Accelerate from Rest to Walk or Jog	●				●	●	●	●	●		●	●	■	■	■	●	●	●	●	●	●
Decelerate from Walk or Jog to Rest	●				●	●		●	●		●	●	■	■	■	●	●	●	●	●	●
Accelerate from Walk to Jog	●				●			●	●		●	●	■	■	■	●	●	●	●	●	●
Decelerate to Walk from Jog	●				●			●	●		●	●	■	■	■	●	●	●	●	●	●
Walk	●				●			●	●		●	●	■	■	■	●	●	●	●	●	●
Jog	●				●			●	●		●	●	■	■	■	●	●	●	●	●	●
Turn in Place (no forward or side movement)	●				●			●	●		●	●	■	■	■	●	●	●	●	●	●
Side Step	●				●			●	●		●	●	■	■	■	●	●	●	●	●	●
Tilt Upper-Body (without Foot Movement)	●				●			●			●	●	■	■	■	●	●	●	●	●	●

Table 9. Absence/presence analysis of LocoX using implementation

	1. Multilimb Coordination	2. Rate Control	3. Response Orientation	4. Reaction Time	5. Speed of Limb Movement	6. Static Strength	7. Explosive Strength	8. Dynamic Strength	9. Trunk Strength	10. Extent Flexibility	11. Dynamic Flexibility	12. Gross Body Coordination	12(A). Side-to-Side Coordination	12(B). Front-to-Back Coordination	12(C). Rotational Coordination	13. Gross Body Equilibrium	13(A). Side-to-Side Equilibrium	13(B). Front-to-Back Equilibrium	13(C). Rotational Equilibrium	14. Stamina	15. Speed of Transport	
Accelerate from Rest to Walk or Jog	4	1	1	1	4	2	4	5	5	1	3	4	■	■	■	■	■	■	■	2	4	
Decelerate from Walk or Jog to Rest	2	1	1	1	2	2	1	2	2	1	2	2	■	■	■	■	2	■	■	■	1	2
Accelerate from Walk to Jog	4	1	1	1	4	1	2	5	5	1	3	4	■	■	■	■	4	■	■	■	2	4
Decelerate to Walk from Jog	5	1	1	1	5	1	1	5	5	1	4	4	■	■	■	■	4	■	■	■	1	4
Walk	4	1	1	1	3	1	1	3	3	1	2	2	■	■	■	■	2	■	■	■	2	2
Jog	5	1	1	1	4	1	1	3	3	1	3	3	■	■	■	■	3	■	■	■	3	2
Turn in Place (no forward or side movement)	4	1	1	1	1	2	1	1	3	2	1	2	■	■	■	■	3	■	■	■	1	1
Side Step	4	1	1	1	1	2	1	1	2	2	1	3	■	■	■	■	3	■	■	■	1	1
Tilt Upper-Body (without Foot Movement)	2	1	1	1	1	3	1	1	5	1	1	3	■	■	■	■	3	■	■	■	1	1

Table 10. Scaled analysis of LocoX using implementation

### 1. Accelerate from Rest to Walk or Jog

In natural locomotion, when the body accelerates from a rest to either a walk or a jog, one of the legs is lifted by tensing the hip flexors and springing off the planted leg using a combination of the quadriceps, glutes and calves. In the walking-in-place motion used for LocoX, all of these muscles are used, however, the stationary leg does not require the glutes to force the thigh back and propel the body forward. LocoX, thus, requires less movement and less overall force and receives 1 point less than natural locomotion in all categories except for explosive strength, in which it receives 2 less points.

### 2. Decelerate from Walk or Jog to Rest

In natural locomotion, deceleration requires a dampening force to stop the body's forward momentum. This force comes mostly from the quadriceps of the forward leg. LocoX does not require such a force as the body is already

stationary. All that is required is for the user to let the lifted leg fall to the vertical position. Thus, the VE device receives only 2 points for all categories.

### **3. Accelerate from Walk to Jog**

Acceleration in natural locomotion requires more force in the same muscles as accelerating from rest to walk or jog does, as well as a greater range of extension of both legs. LocoX also requires a similar increase in force and range of extension, however, there is no need to propel the body forward, and thus, the glutes of the stationary leg are not utilized as much. For this, LocoX receives 1 point less than natural locomotion in all categories, except for explosive strength, in which it receives 2 less points.

### **4. Decelerate to Walk from Jog**

This motion is similar to walk or jog to rest but requires less strength. LocoX is similar to natural locomotion in this regard and receives the same score as natural locomotion in all categories.

### **5. Walk**

LocoX only requires the use of hip flexors for walking motion to lift each leg in succession. The stationary leg requires no backswing motion to propel the body forward. Thus, LocoX receives 1 point less than natural locomotion in all categories.

### **6. Jog**

LocoX requires the exact same muscles as natural locomotion for this motion except for the need to propel the body forward, and thus, receives 1 point less than natural locomotion in all categories.

### **7. Turn in Place (no forward or side movement)**

LocoX uses the same motion and receives the same points as natural locomotion.

## **8. Sidestep**

Sidestepping in LocoX only requires a single step to the side as opposed to repeated steps and thus only receives 2 points for all categories.

## **9. Tilt Upper-Body (without foot movement)**

The same as natural locomotion in all categories.

## **B. TEST RESULTS**

In initial tests using LocoX in America's Army while online, it demonstrated a 90% recognition rate for walking forward, sidestepping and backstepping. 100% of the recognition failures were transitions into the sidestep movement state. Depending upon a person's gait, LocoX had difficulty determining which side, left or right, the sidestep was toward. Improvements to the sidestep conditionals by identifying the side of the leg with the highest velocity as being the intended direction, however, improved the rate to a 95% recognition rate in latter tests, but those tests did not contain a large enough test subject group to eliminate familiarity with the device as the factor.

In LocoX, the DO:DI ratio, decreased from the initial goal of 11.5 to 6 due to not implementing crouching, crawling, an jumping. The only movements implemented were walking forward, sidestepping left or right, back stepping, running forward, and jumping. The DO:DI ratio still exceeds the 2.78 ratio of the mouse and keyboard. The remaining movements were not implemented only due to time constraints. The possible combinations of positions and velocities of the upper legs contains untapped cues for crouching and even crawling, the former state reached when both legs are in the forward position, and the latter when both legs are in the backward position.

Although the recognition of the motions was exceptionally accurate, however, the implementation was unable to work in any computer besides a PC with an AMD Athlon XP processor on a Tyan brand motherboard due to the low level coding used to simulate keystrokes for use in a DOS based PC game,

America's Army. Although the movements were still recognized by the algorithm, the keypresses could not be recreated correctly on any other computer.

### C. CONCLUSION

In this chapter, we have demonstrated the walking-in-place motion used in LocoX to be very similar in specific muscles required and the force necessary to the ODT, but without the additional requirements for equilibrium that transitions on the ODT require. Furthermore, the walking-in-place motion is something that everyone already knows how to do, and thus, is a very natural metaphor for locomotion. However, both sidestepping and back stepping present a situation where the walking-in-place motion must be augmented with an unnatural metaphor, that of stepping to the side or backwards without any continuous movement while the user viewpoint continues to move in the virtual environment, as if the user is sliding. Because the initial movement is natural and feedback in the intended motion is quickly received, the naturalness is intact until movement continues without any extra action on the user's behalf. Still, in most cases, the walking-in-place motion does not require the same balance required in the ODT, and can offer additional degrees of locomotion output. By this comparison, LocoX compares favorably to the ODT with considerably less footprint and cost.

## **VI. CONCLUSION AND FUTURE WORK**

### **A. CONCLUSION**

We have seen how past taxonomies of locomotion failed to place enough emphasis on the human component of human interaction, instead, focusing on device-centric cues as limited by current technology. As a result, developers of locomotion devices have been systemically unaware of the wide range of cues available to them in developing locomotion devices. By understanding the range of cues available, which the proposed taxonomy outlines, researchers will be able to make the conscious decision of specific cues to explore the effectiveness of. As research in those areas continues to build, quantitative comparisons based on Fitts' law or F-JAS will enable future designers to choose which limb they wish to use in an interaction with a relatively confident understanding of how it will perform for a given function.

The actual implementation developed in this work provides an initial look at the type of locomotion device that can be designed by making conscious decisions about what level of cue to use, what limb to take that cue from, and how many functions to map the limb to. The degree of functionality in this work exceeded those of many past devices with much more information from less sensors, a smaller footprint, and much lower associated costs.

### **B. FUTURE WORK**

While initial results from the taxonomy and locomotion device appear promising, the full domain of cues from the human body remains undiscovered. In addition, limitations in the implemented locomotion device, due to low level coding providing different outputs based on the motherboard, processor and operating system used, created difficulty in the quantitative testing of LocoX. The following lists issues that remain for future work:

- Further improve the detection of lateral movements
- Decouple movement direction from viewpoint direction in a specialized simulator using a head mounted display
- Debug simulated keystroke messages for use on PCs other than the one used to develop the original implementation so that the device can be used in any DirectX game
- Further divide orders of immediacy into distinct muscle groupings and commence testing various interaction methods using each grouping to determine the exact relationship between ease of use and levels of articulation in interaction devices

## APPENDIX A. TRANSITION MATRICES

**Move:** Stand

P	V	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	OO	00	↑O	↓O	←O	→O	O↑	O↓	O→	O←	↑↑	↑↓	↓↑	↓↓	↔	↔	↔	
0	OO	0	0	0	0		0	0	0		0	0	0	0				
1	↑O		1								1	1						
2	↓O			3									3	3				
3	←O				5				5						5	7	5	
4	O↑					1					1		1					
5	O↓						3					3		3				
6	O→				7				7						7	7	5	
7	↑↑																	
8	↑↓											3						
9	↓↑												3					
10	↓↓																	
11	↔				5				7									

**Move: Fwd Upswing**

P	V	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	OO	00	↑○	↓○	←○	→○	○↑	○↓	○→	○←	↑↑	↑↓	↓↑	↓↓	↔	↔	→→	→←
0	OO	0	0	0	0		0	0	0		0	0	0	0				
1	↑○	1	1	2			1	1			1	1	2	2				
2	↓○			3									3	3				
3	←○				5				5						5	7	5	
4	○↑	1	1	1			1	2			1	2	1	2				
5	○↓						3				3		3					
6	○→				7			7							7	7	5	
7	↑↑																	
8	↑↓										3							
9	↓↑											3						
10	↓↓																	
11	↔				5				7									

**Move: Fwd Downswing**

P	V	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	OO	00	↑○	↓○	←○	→○	○↑	○↓	○→	○←	↑↑	↑↓	↓↑	↓↓	↔	↔	→→	→←
0	OO	0	0	0	0		0	0	0		0	0	0	0				
1	↑○	2	1	2			2	2			1	1	2	2				
2	↓○			3									3	3				
3	←○				5				5						5	7	5	
4	○↑	2	2	2			1	2			1	2	1	2				
5	○↓						3				3		3					
6	○→				7				7						7	7	5	
7	↑↑																	
8	↑↓									3								
9	↓↑										3							
10	↓↓																	
11	↔				5				7									

**Move: Back Upswing**

P	V	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
P	OO	00	↑○	↓○	←○	→○	○↑	○↓	○→	○←	↑↑	↑↓	↓↑	↓↓	↔	↔	→→	→←
0	OO	0	0	0	0		0	0	0		0	0	0	0				
1	↑○																	
2	↓○	3	3	3			3	3			3	3	3	3				
3	←○				5				5						5	7	5	
4	○↑																	
5	○↓	3	3	3			3	3			3	3	3	3				
6	○→				7			7							7	7	5	
7	↑↑																	
8	↑↓										3							
9	↓↑											3						
10	↓↓																	
11	↔				5				7									

**Move: Back Downswing**

P	V	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
P	OO	00	↑○	↓○	←○	→○	○↑	○↓	○→	○←	↑↑	↑↓	↓↑	↓↓	↔	↔	→→	→←
0	OO	0	0	0	0		0	0	0		0	0	0	0				
1	↑○																	
2	↓○	3	3	3			3	3			3	3	3	3				
3	←○				5				5						5	7	5	
4	○↑																	
5	○↓	3	3	3			3	3			3	3	3	3				
6	○→				7			7							7	7	5	
7	↑↑																	
8	↑↓										3							
9	↓↑											3						
10	↓↓																	
11	↔				5				7									

**Move: Left Upswing**

P	V	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
P	OO	OO	↑O	↓O	←O	→O	O↑	O↓	O→	O←	↑↑	↑↓	↓↑	↓↓	↔	↔	→	→
0	OO	5	0	0	0		0	0	0		0	0	0	0				
1	↑O		1								1	1						
2	↓O	3	3	3			3	3			3	3	3	3				
3	←O				5	6			5	6					5	6	5	6
4	O↑					1					1		1					
5	O↓	3	3	3			3	3			3	3	3	3				
6	O→				6	5			6	5					5	6	5	5
7	↑↑																	
8	↑↓										3							
9	↓↑											3						
10	↓↓																	
11	↔				5	6			5	6					5	5	5	6

**Move: Left Downswing**

P	V	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
P	OO	OO	↑O	↓O	←O	→O	O↑	O↓	O→	O←	↑↑	↑↓	↓↑	↓↓	↔	↔	→	→
0	OO	0	0	0	0		0	0	0		0	0	0	0				
1	↑O		1								1	1						
2	↓O	3	3	3			3	3			3	3	3	3				
3	←O				5	6			5	6					5	6	5	6
4	O↑					1					1		1					
5	O↓	3	3	3			3	3			3	3	3	3				
6	O→				6	5			6	5					5	6	5	5
7	↑↑																	
8	↑↓										3							
9	↓↑											3						
10	↓↓																	
11	↔				5	6			5	6					5	0	5	6

**Move: Right Upswing**

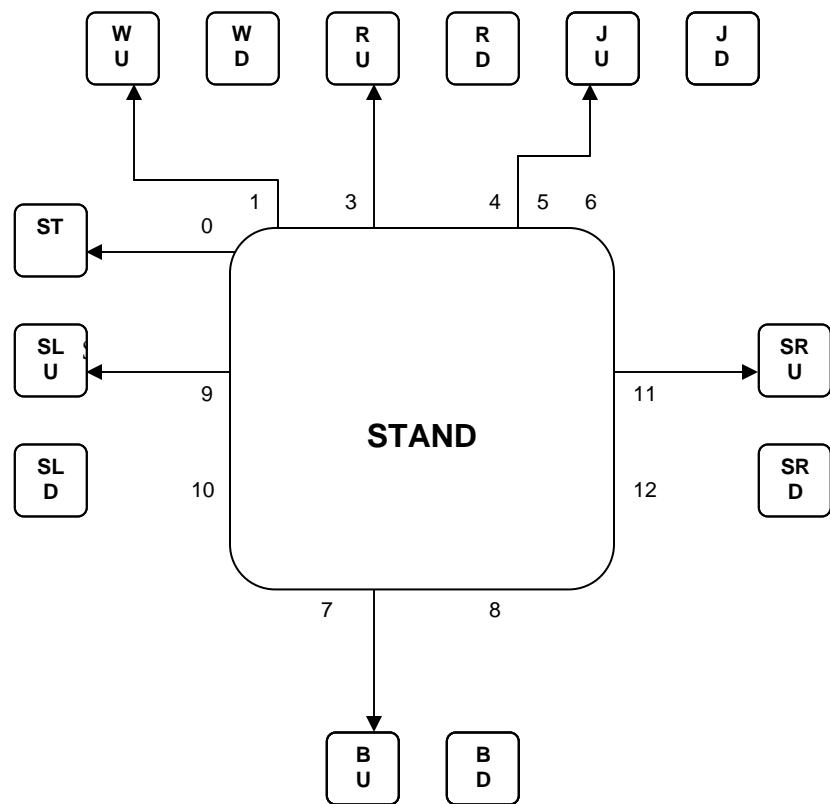
P	V	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
P	OO	00	↑○	↓○	←○	→○	○↑	○↓	○→	○←	↑↑	↑↓	↓↑	↓↓	↔	↔	→	→
0	OO	7	0	0	0		0	0	0		0	0	0	0				
1	↑○		1								1	1						
2	↓○	3	3	3			3	3			3	3	3	3				
3	←○				8	7			8	7					7	7	8	8
4	○↑						1				1		1					
5	○↓	3	3	3			3	3			3	3	3	3				
6	○→				7	8			7	8					7	7	8	8
7	↑↑																	
8	↑↓										3							
9	↓↑											3						
10	↓↓																	
11	↔				7	8			7	8					7	7	7	8

**Move: Right Downswing**

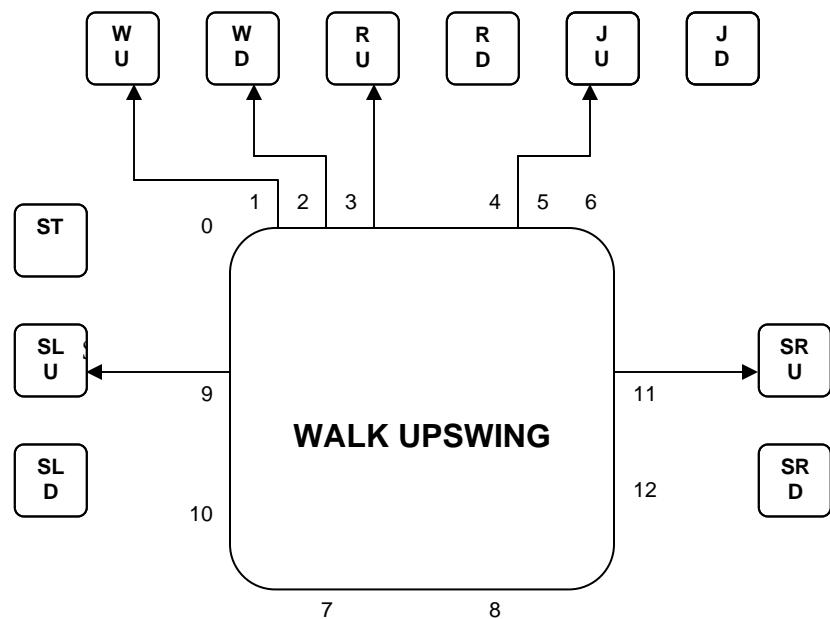
P	V	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
P	OO	00	↑○	↓○	←○	→○	○↑	○↓	○→	○←	↑↑	↑↓	↓↑	↓↓	↔	↔	→	→
0	OO	0	0	0	0		0	0	0		0	0	0	0				
1	↑○		1								1	1						
2	↓○	3	3	3			3	3			3	3	3	3				
3	←○				8	7			8	7					7	7	8	8
4	○↑						1				1		1					
5	○↓	3	3	3			3	3			3	3	3	3				
6	○→				7	8			7	8					7	7	8	8
7	↑↑																	
8	↑↓										3							
9	↓↑											3						
10	↓↓																	
11	↔				7	8			7	8					7	7	0	8

THIS PAGE INTENTIONALLY LEFT BLANK

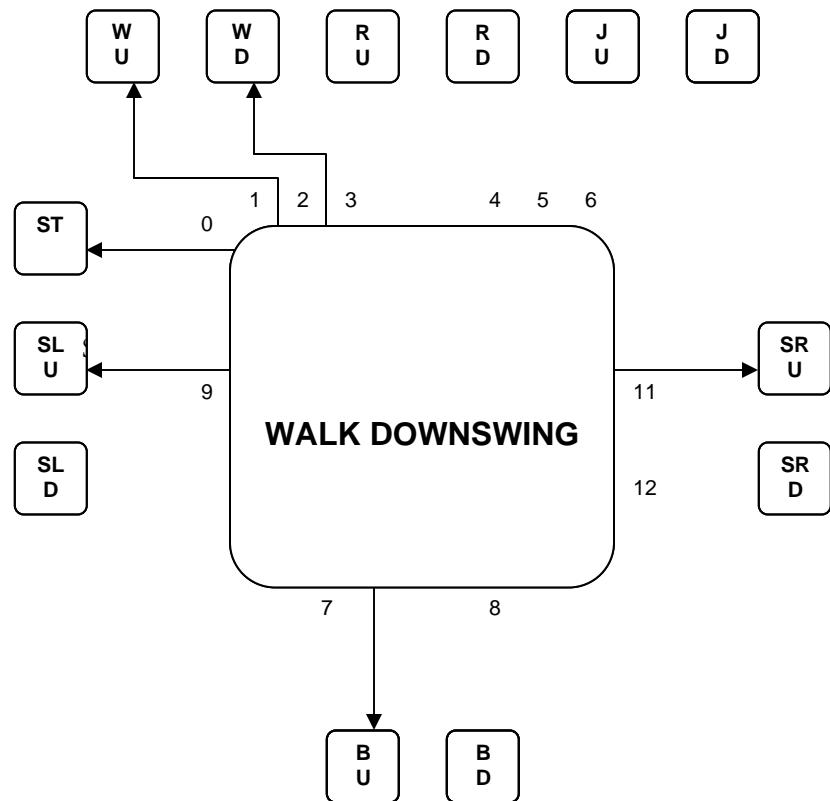
## APPENDIX B. TRANSITION DIAGRAM NOTES



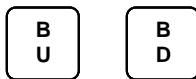
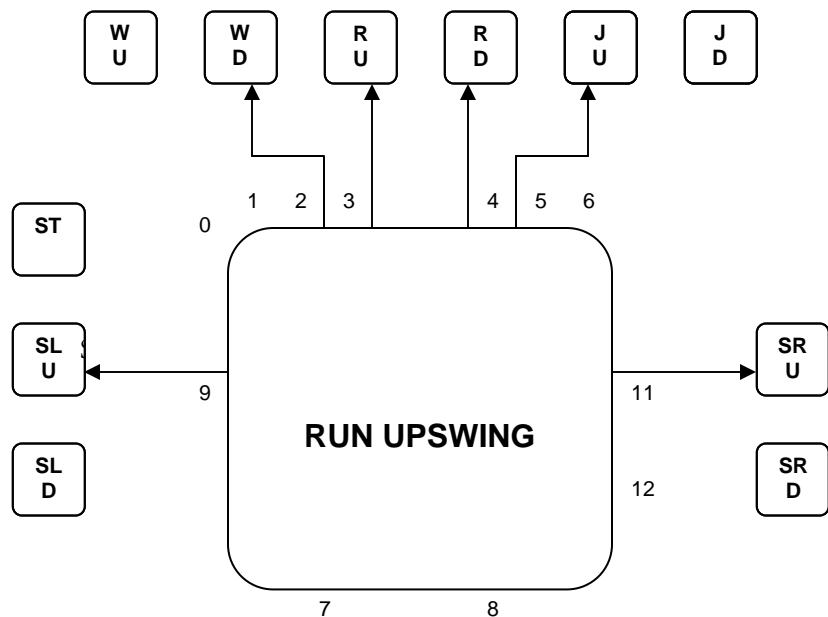
1. Can only transition to upswing movements from stand.



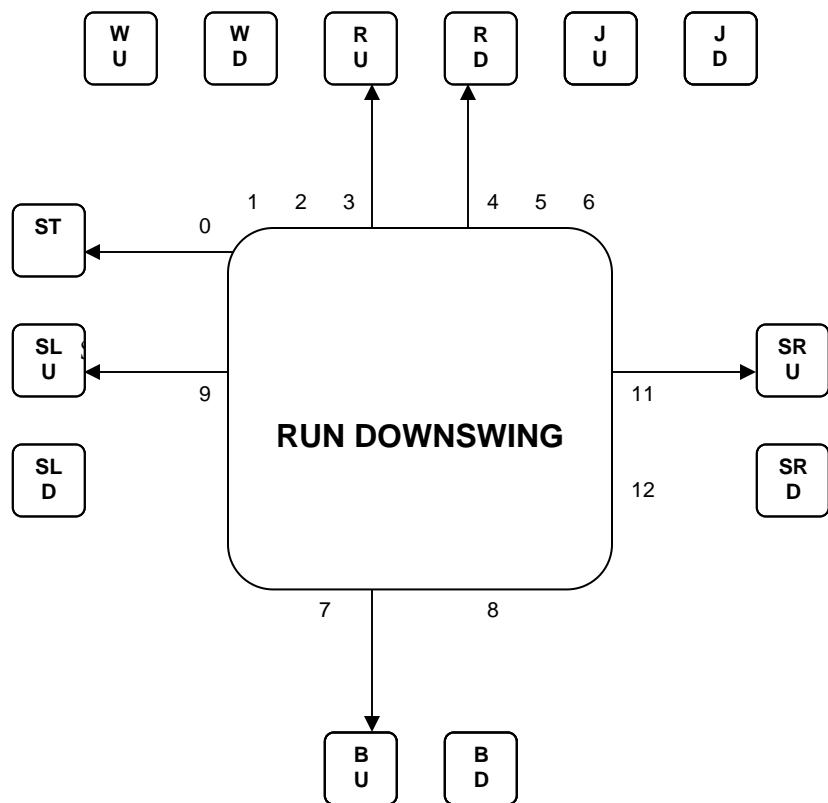
1. Cannot transition to backstep upswing before going to walk downswing.
2. All other upswing transitions are possible.
3. Cannot transition to stand until transition to walk downswing.
4. No other downswing movements besides walk are possible.



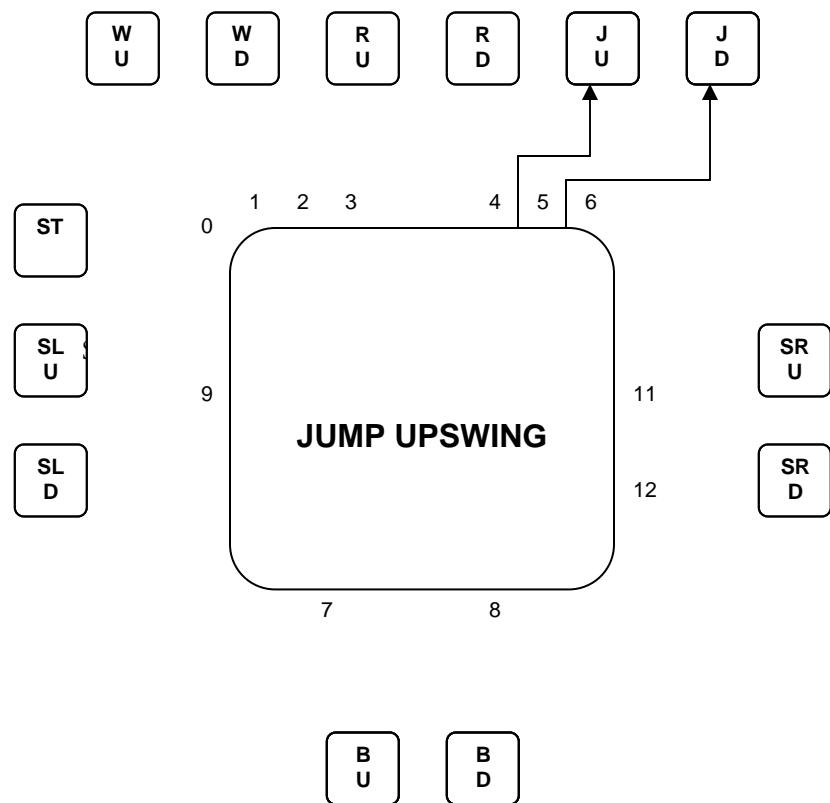
1. Can now transition to stand.
2. All upswing movements possible except for run and jump.



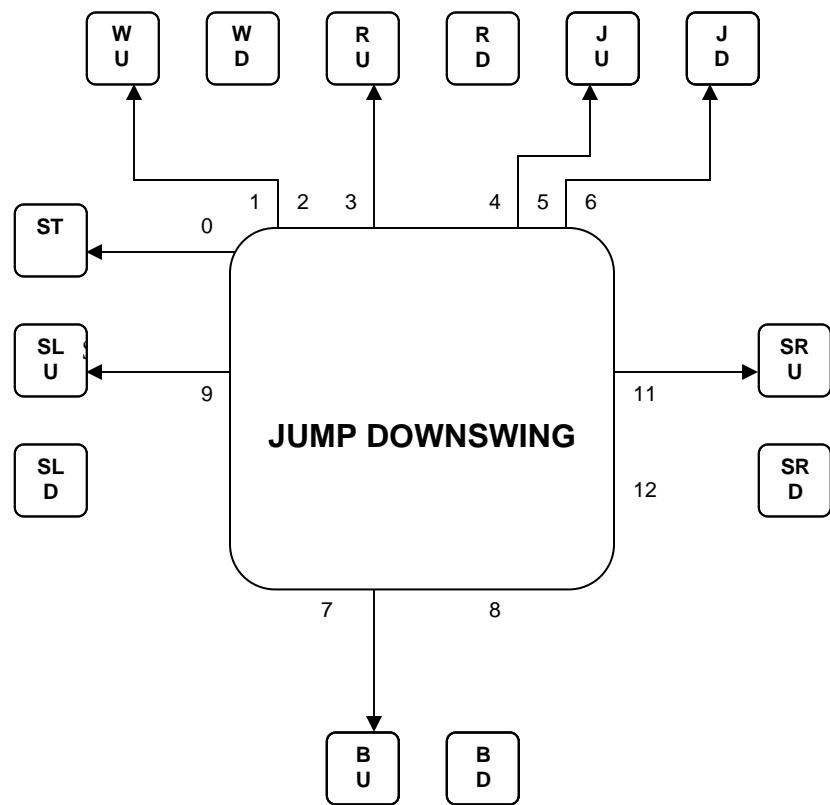
1. Cannot transition to backstep upswing from here.
2. Can transition to walk upswing if during run upswing, begin a downswing before reaching the run threshold.
3. Cannot transition to stand until go to run downswing.



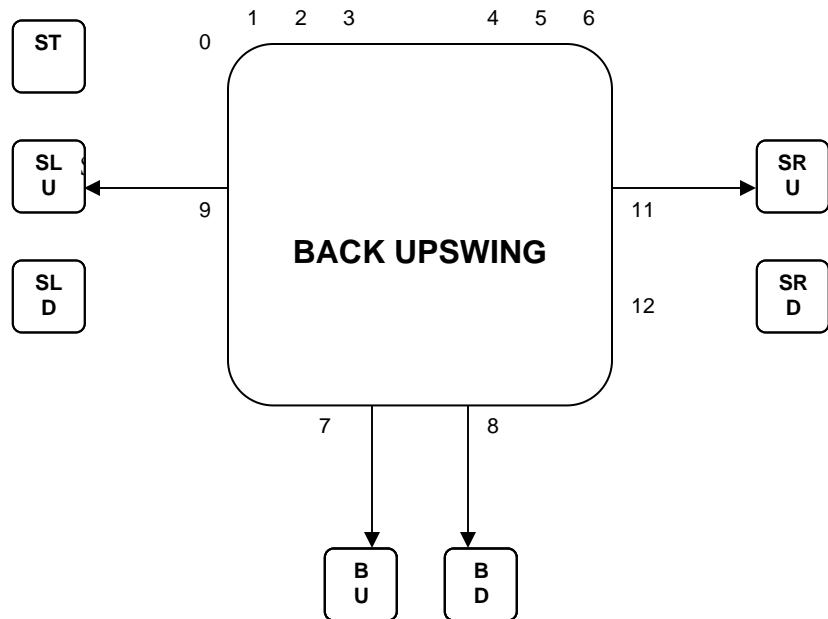
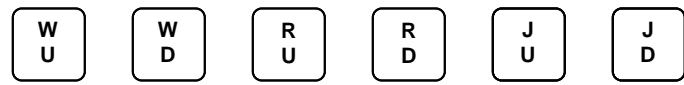
1. Can now transition to stand.
2. All other upswing movements possible except walk upswing.



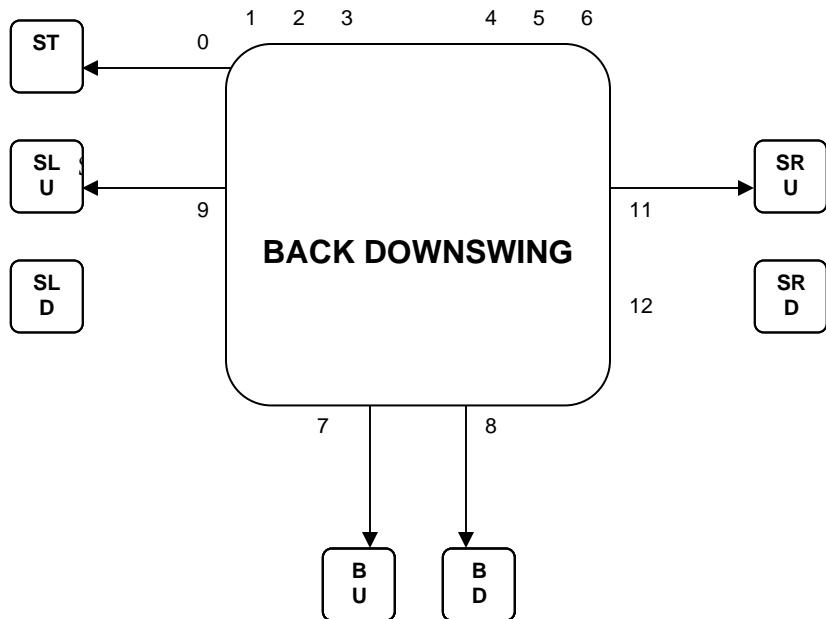
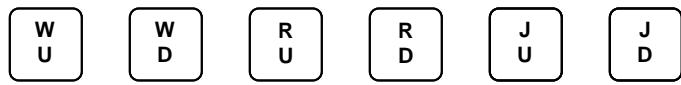
1. No movement transitions are possible when jumping except jump downswing.



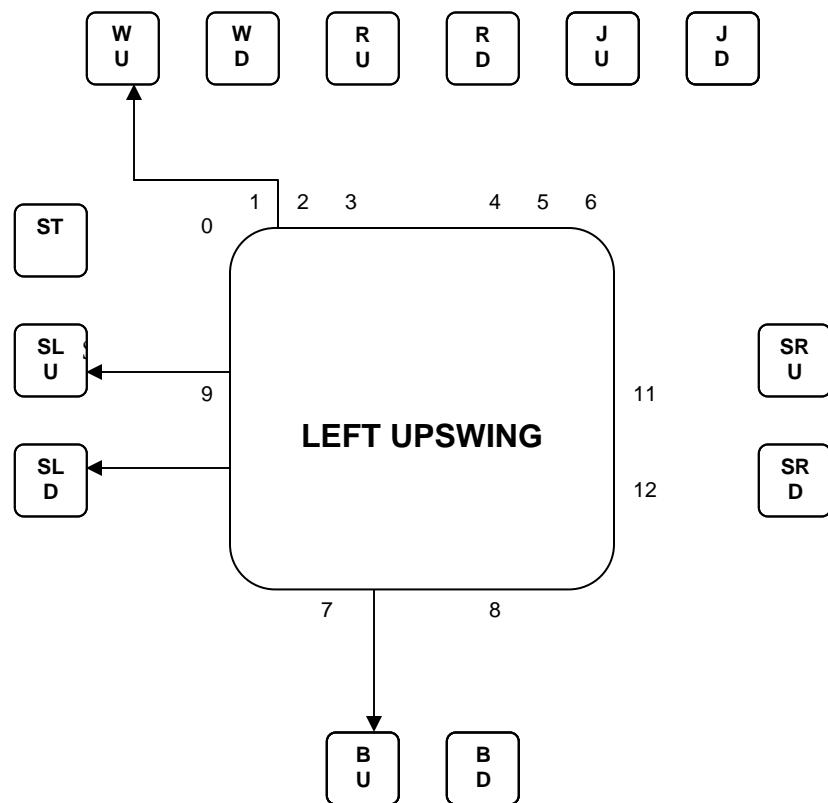
- Upon downswing, all upward movements are possible.



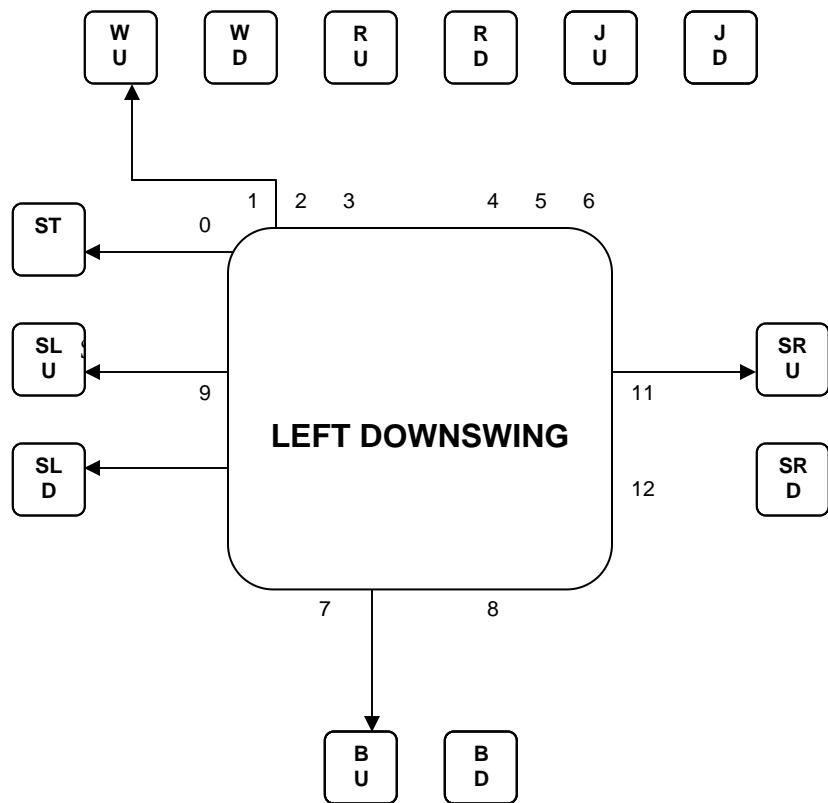
1. No forward movements possible when in this state. Only sidestep movements possible, just as in walk upswing where sidesteps possible, but no backstep.



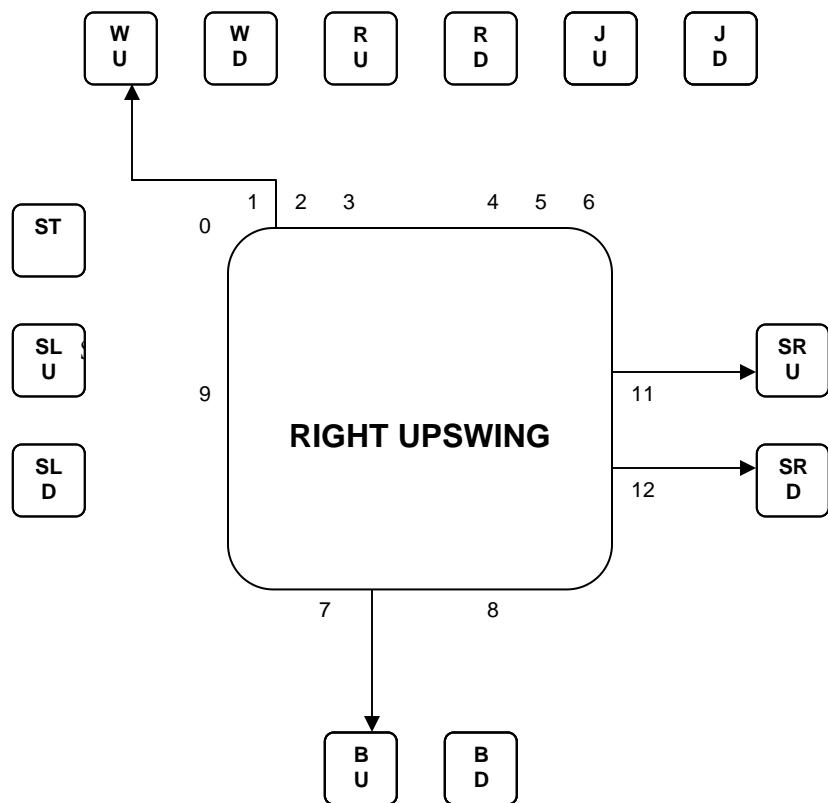
1. Cannot transition to forward movements unless in stand state first.



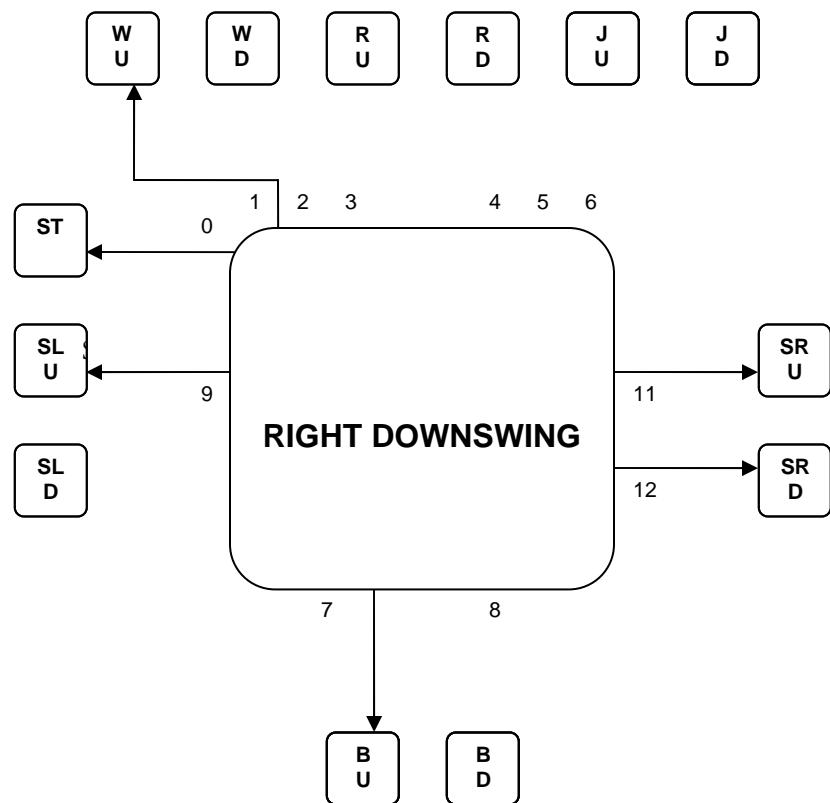
1. Sidestep right movements not possible until sidestep left downswing.



1. Now that in left downswing, other transitions possible, including sidestep right upswing.



1. Sidestep left movements not possible until sidestep right downswing.



1. Now that in right downswing, other transitions possible, including sidestep left upswing.

THIS PAGE INTENTIONALLY LEFT BLANK

## APPENDIX C. C++ IMPLEMENTATION OF ALGORITHM

```
//////////  
//  
// LocoEngine.cpp: implementation of the LocoEngine class.  
//  
//////////  
  
#include "isense.h"  
#include "LocoEngine.h"  
#include <math.h>  
#include <time.h>  
#include <conio.h>  
#include <winable.h>  
#include <windows.h>  
#include <iostream>  
#include <iomanip>  
#include <dos.h>  
#include <stdio.h>  
  
using namespace std;  
  
//////////  
//  
// Construction/Destruction  
//  
//////////  
LocoEngine::LocoEngine(LOCODATA_TYPE* d)  
{  
    dataPtr = d;  
  
    // Old data  
    old_lh = 0;  
    old_lp = 0;  
    old_lr = 0;  
    old_rh = 0;  
    old_rp = 0;  
    old_rr = 0;  
    old_move = 0;  
  
    // Time  
    old_time = 0;  
    time = double (clock()) / 1000;  
  
    // Analysis  
    left_posit = 0;  
    right_posit = 0;  
    left_veloc = 0;  
    right_veloc = 0;  
  
    // Parameters  
    position = 0;  
    velocity = 0;  
    cycle = 0;  
}  
  
LocoEngine::~LocoEngine()  
{  
}  
  
//////////  
//  
// getMove  
//
```

```

// Determine next move based on heading, pitch and roll data for each leg.
//
void LocoEngine::getMove()
{
    old_time = time;
    old_move = dataPtr->move;
    time = double (clock()) / 1000;

    deriveVelocityData();
    findPosition();
    findVelocity();
    findMove();
    saveOldData();
}

///////////////////////////////
// 
// deriveVelocityData
// 
// Derive velocity from old and new position data.
// 
void LocoEngine::deriveVelocityData()
{
    // Left leg
    dataPtr->lhd = (dataPtr->lh - old_lh) / (time - old_time);
    dataPtr->ldp = (dataPtr->lp - old_lp) / (time - old_time);
    dataPtr->ldr = (dataPtr->lr - old_lr) / (time - old_time);

    // Right leg
    dataPtr->rdh = (dataPtr->rh - old_rh) / (time - old_time);
    dataPtr->rdp = (dataPtr->rp - old_rp) / (time - old_time);
    dataPtr->rdr = (dataPtr->rr - old_rr) / (time - old_time);

    cout << fixed << setprecision(2)
        << setw(3) << "lp:" << setw(8) << dataPtr->lp
        << setw(1) << ' '
        << setw(3) << "rp:" << setw(8) << dataPtr->rp
        << setw(1) << ' '
        << setw(3) << "lr:" << setw(8) << dataPtr->lr
        << setw(1) << ' '
        << setw(3) << "rr:" << setw(8) << dataPtr->rr
        //<< setw(1) << ' '
        //<< setw(4) << "lhd:" << setw(8) << dataPtr->lhd
        //<< setw(1) << ' '
        //<< setw(4) << "ldp:" << setw(8) << dataPtr->ldp
        //<< setw(1) << ' '
        //<< setw(4) << "rdp:" << setw(8) << dataPtr->rdp
        //<< setw(1) << ' '
        //<< setw(4) << "ldr:" << setw(8) << dataPtr->ldr
        //<< setw(1) << ' '
        //<< setw(4) << "rdr:" << setw(8) << dataPtr->rdr
        << setw(1) << ' ' << endl;
}

///////////////////////////////
// 
// findPosition
// 
// Determine position situation of legs.
// 
void LocoEngine::findPosition()
{
    left_posit          = 0;
    right_posit         = 0;
}

```

```

const double PR_PRATIO = 1;           // Sensitivity ratio between pitch and roll
const double F1_THRESH = 10;          // Front 1 threshold
const double F2_THRESH = 30;          // Front 2 threshold
const double F3_THRESH = 90;          // Front 3 threshold
const double B1_THRESH = -10;         // Back 1 threshold
const double S1_THRESH = 5;           // Side 1 threshold

const int positionArray[6][6] = {

    { 0, 6, 7, 8, 9, 10 },
    { 1, 15, 17, 17, 11, 0 },
    { 2, 16, 18, 18, 12, 0 },
    { 3, 16, 18, 18, 12, 0 },
    { 4, 13, 14, 14, 19, 0 },
    { 5, 0, 0, 0, 0, 20 }
};

///////////////////////////////
// Analyze left leg position
///////////////////////////////

// If left leg position in front quadrant
if (PR_PRATIO * dataPtr->lp >= dataPtr->lr &&
    PR_PRATIO * dataPtr->lp >= -dataPtr->lr) {

    // and left leg pitch exceeds front 3 threshold, assign front 3 position
    if (dataPtr->lp > F3_THRESH) {
        left_posit = 3;
    }

    // and left leg pitch exceeds front 2 threshold, assign front 2 position
    else if (dataPtr->lp > F2_THRESH) {
        left_posit = 2;
    }

    // and left leg pitch exceeds front 1 threshold, assign front 1 position
    else if (dataPtr->lp > F1_THRESH) {
        left_posit = 1;
    }
}

// If left leg position in rear quadrant
if (PR_PRATIO * dataPtr->lp <= -dataPtr->lr &&
    PR_PRATIO * dataPtr->lp <= dataPtr->lr) {

    // and left leg pitch exceeds back 1 threshold, assign back 1 position
    if (dataPtr->lp < B1_THRESH) {
        left_posit = 4;
    }
}

// If left leg position in left quadrant
if (PR_PRATIO * dataPtr->lp < dataPtr->lr &&
    PR_PRATIO * dataPtr->lp > -dataPtr->lr) {

    // and right leg roll exceeds sidestep 1 threshold, assign left 1 position
    if (dataPtr->lr > S1_THRESH) {
        left_posit = 5;
    }
}

///////////////////////////////
// Analyze right leg position
///////////////////////////////

```

```

// If right leg position in front quadrant
if (PR_PRATIO * dataPtr->rp >= dataPtr->rr &&
    PR_PRATIO * dataPtr->rp >= -dataPtr->rr) {

    // and right leg pitch exceeds front 3 threshold, assign front 3 position
    if (dataPtr->rp > F3_THRESH) {
        right_posit = 3;
    }

    // and right leg pitch exceeds front 2 threshold, assign front 2 position
    else if (dataPtr->rp > F2_THRESH) {
        right_posit = 2;
    }

    // and right leg pitch exceeds front 1 threshold, assign front 1 position
    else if (dataPtr->rp > F1_THRESH) {
        right_posit = 1;
    }
}

// If right leg position in rear quadrant
if (PR_PRATIO * dataPtr->rp <= -dataPtr->rr &&
    PR_PRATIO * dataPtr->rp <= dataPtr->rr) {

    // and right leg roll exceeds back 1 threshold, assign back 1 position
    if (dataPtr->rp < B1_THRESH) {
        right_posit = 4;
    }
}

// If right leg position in right quadrant
if (PR_PRATIO * dataPtr->rp < -dataPtr->rr &&
    PR_PRATIO * dataPtr->rp > dataPtr->rr) {

    // and right leg roll exceeds sidestep 1 threshold, assign right 1 position
    if (dataPtr->rr < -S1_THRESH) {
        right_posit = 5;
    }
}

///////////////////////////////
// Determine position situation
// ///////////////////////
position = positionArray[left_posit][right_posit];
/*
cout << fixed << setprecision(2)
    << setw(5) << "lpos:" << setw(2) << left_posit
    << setw(1) << ' '
    << setw(5) << "rpos:" << setw(2) << right_posit
    << setw(1) << ' '
    << setw(2) << "P:"    << setw(2) << position
    << setw(2) << ' ';
*/
///////////////////////////////
// findVelocity
// /////////////////////
// Determine velocity situation of legs.
// /////////////////////
void LocoEngine::findVelocity()
{
    left_veloc                  = 0;
    right_veloc                 = 0;
}

```

```

const double PR_VRATIO      = 2; // Sensitivity ratio between pitch and roll
const double FRONTAL_THRESH = 10; // Frontal velocity threshold
const double SAGITTAL_THRESH = 5; // Sagittal velocity threshold

const int velocityArray[5][5] = {

    { 0, 5, 6, 7, 8 },
    { 1, 9, 11, 0, 0 },
    { 2, 12, 10, 0, 0 },
    { 3, 0, 0, 13, 15 },
    { 4, 0, 0, 16, 14 }
};

///////////////////////////////
// If both legs are straight, decide if frontal or sagittal velocity dominant
// and determine velocity situation.
//
/////////////////////////////
if (position == 0) {

    ///////////////////////////////
    //
    // Analyze left leg.
    //
    //////////////////////////////

    // If left leg pitch velocity greater than roll velocity in forward direction
    if (PR_VRATIO * dataPtr->ldp >= dataPtr->ldr &&
        PR_VRATIO * dataPtr->ldp >= -dataPtr->ldr &&
        dataPtr->ldp > FRONTAL_THRESH) {

        // then left leg velocity is frontal w/ forward motion
        left_veloc = 1;
    }

    // If left leg pitch velocity greater than roll velocity in backward direction
    else if (PR_VRATIO * dataPtr->ldp <= -dataPtr->ldr &&
              PR_VRATIO * dataPtr->ldp <= dataPtr->ldr &&
              dataPtr->ldp < -FRONTAL_THRESH) {

        // then left leg velocity is frontal w/ backward motion
        left_veloc = 2;
    }

    // If left leg pitch velocity less than roll velocity in left direction
    else if (PR_VRATIO * dataPtr->ldp < dataPtr->ldr &&
              PR_VRATIO * dataPtr->ldp > -dataPtr->ldr &&
              dataPtr->ldr > SAGITTAL_THRESH) {

        // then left leg velocity is sagittal w/ outward motion
        left_veloc = 3;
    }

    // If left leg pitch velocity less than roll velocity in right direction
    else if (PR_VRATIO * dataPtr->ldp < -dataPtr->ldr &&
              PR_VRATIO * dataPtr->ldp > dataPtr->ldr &&
              dataPtr->ldr < -SAGITTAL_THRESH) {

        // then left leg velocity is sagittal w/ inward motion
        left_veloc = 4;
    }

    ///////////////////////////////
    //
    // Analyze right leg.
    //
    //////////////////////////////
}

```

```

// If right leg pitch velocity greater than roll velocity in forward direction
if (PR_VRATIO * dataPtr->rdp >= dataPtr->rdr &&
    PR_VRATIO * dataPtr->rdp >= -dataPtr->rdr &&
    dataPtr->rdp > FRONTAL_THRESH) {

    // then right leg velocity is frontal w/ forward motion
    right_veloc = 1;
}

// If right leg pitch velocity greater than roll velocity in backward direction
else if (PR_VRATIO * dataPtr->rdp <= -dataPtr->rdr &&
         PR_VRATIO * dataPtr->rdp <= dataPtr->rdr &&
         dataPtr->rdp < -FRONTAL_THRESH) {

    // then right leg velocity is frontal w/ backward motion
    right_veloc = 2;
}

// If right leg pitch velocity less than roll velocity in right direction
else if (PR_VRATIO * dataPtr->rdp < -dataPtr->rdr &&
         PR_VRATIO * dataPtr->rdp > dataPtr->rdr &&
         dataPtr->rdr < -SAGITTAL_THRESH) {

    // then right leg velocity is sagittal w/ outward motion
    right_veloc = 3;
}

// If right leg pitch velocity less than roll velocity in left direction
else if (PR_VRATIO * dataPtr->rdp < dataPtr->rdr &&
         PR_VRATIO * dataPtr->rdp > -dataPtr->rdr &&
         dataPtr->rdr > SAGITTAL_THRESH) {

    // then right leg velocity is sagittal w/ inward motion
    right_veloc = 4;
}

// If both legs are either fwd or back, only frontal velocities possible.
// Analyze right leg.
// If right leg velocity is forward
if (dataPtr->rdp > FRONTAL_THRESH) {

```

---

```

// If both legs are either fwd or back, only frontal velocities possible.
// Analyze right leg.
// If right leg velocity is forward
if (dataPtr->rdp > FRONTAL_THRESH) {

```

```

        // then left leg velocity is frontal w/ forward motion
        left_veloc = 1;
    }

    // If left leg velocity is backward
    else if (dataPtr->ldp < -FRONTAL_THRESH) {

        // then left leg velocity is frontal w/ backward motion
        left_veloc = 2;
    }

    ///////////////////////////////////////////////////////////////////
    //
    // Analyze right leg.
    //
    ///////////////////////////////////////////////////////////////////

    // If right leg velocity is forward
    if (dataPtr->rdp > FRONTAL_THRESH) {

        // then right leg velocity is frontal w/ forward motion
        right_veloc = 1;
    }

    // If right leg velocity is backward
    else if (dataPtr->rdp < -FRONTAL_THRESH) {

        // then right leg velocity is frontal w/ backward motion
        right_veloc = 2;
    }
}

///////////////////////////////////////////////////////////////////
//
// If both legs are either left or right, only sagittal velocities possible.
//
///////////////////////////////////////////////////////////////////
if (position == 5 ||
    position == 10 ||
    position == 20) {

    ///////////////////////////////////////////////////////////////////
    //
    // Analyze left leg velocity.
    //
    ///////////////////////////////////////////////////////////////////

    // If left leg velocity is left
    if (dataPtr->ldr > SAGITTAL_THRESH) {

        // then left leg velocity is sagittal w/ outward motion
        left_veloc = 3;
    }

    // If left leg velocity is right
    else if (dataPtr->ldr < -SAGITTAL_THRESH) {

        // then left leg velocity is sagittal w/ inward motion
        left_veloc = 4;
    }

    ///////////////////////////////////////////////////////////////////
    //
    // Analyze right leg.
    //
    ///////////////////////////////////////////////////////////////////

    // If right leg velocity is right

```

```

        if (dataPtr->rdr < -SAGITTAL_THRESH) {

            // then right leg velocity is sagittal w/ outward motion
            right_veloc = 3;
        }

        // If right leg velocity is left
        else if (dataPtr->rdr > SAGITTAL_THRESH) {

            // then right leg velocity is sagittal w/ inward motion
            right_veloc = 4;
        }
    }

///////////////////////////////
// Simplify frontal velocity situations by converting situations 11 & 12
// into situations 1, 2, 5, or 6.
//
///////////////////////////////

// If left leg velocity is forward and right leg velocity is backward
if (left_veloc == 1 && right_veloc == 2) {

    // and left leg velocity is greater than right leg velocity
    if (dataPtr->ldp >= -dataPtr->rdp) {

        // then set right leg velocity to zero
        right_veloc = 0;
    }

    // and right leg velocity is greater than left leg velocity
    else if (dataPtr->ldp < -dataPtr->rdp) {

        // then set left leg velocity to zero
        left_veloc = 0;
    }
}

// If left leg velocity is backward and right leg velocity is forward
else if (left_veloc == 2 && right_veloc == 1) {

    // and left leg velocity is greater than right leg velocity
    if (-dataPtr->ldp > dataPtr->rdp) {

        // then set right leg velocity to zero
        right_veloc = 0;
    }

    // and right leg velocity is greater than left leg velocity
    else if (-dataPtr->ldp <= dataPtr->rdp) {

        // then set left leg velocity to zero
        left_veloc = 0;
    }
}

///////////////////////////////
// Simplify sagittal velocity situations by converting situations 13 thru 16
// into situations 0, 3, 4, 7, or 8.
//
///////////////////////////////

// If left leg and right leg velocity are outward
if (left_veloc == 3 && right_veloc == 3) {

    // and left leg velocity is greater than right leg velocity
    if (dataPtr->ldr > -dataPtr->rdr) {

```

```

        // then set right leg velocity to zero
        right_veloc = 0;
    }

    // and right leg velocity is greater than left leg velocity
    else if (dataPtr->ldr < -dataPtr->rdr) {

        // then set left leg velocity to zero
        left_veloc = 0;
    }
}

// If left leg and right leg velocity are inward
else if (left_veloc == 4 && right_veloc == 4) {

    // and left leg velocity is greater than right leg velocity
    if (-dataPtr->ldr > dataPtr->rdr) {

        // then set right leg velocity to zero
        right_veloc = 0;
    }

    // and right leg velocity is greater than left leg velocity
    else if (-dataPtr->ldr < dataPtr->rdr) {

        // then set left leg velocity to zero
        left_veloc = 0;
    }
}

// If left leg and right leg velocity are left
if (left_veloc == 3 && right_veloc == 4) {

    // then set right leg velocity to zero
    right_veloc = 0;
}

// If left leg and right leg velocity are right
if (left_veloc == 4 && right_veloc == 3) {

    // then set left leg velocity to zero
    left_veloc = 0;
}

velocity = velocityArray[left_veloc][right_veloc];
/*
cout << fixed << setprecision(2)
    << setw(3) << "lv:" << setw(2) << left_veloc
    << setw(1) << ' '
    << setw(3) << "rv:" << setw(2) << right_veloc
    << setw(1) << ' '
    << setw(2) << "V:"  << setw(2) << velocity
    << setw(1) << ' ';
*/
}

///////////////////////////////
//
// findMove
//
// Determine state transition from new data.
//
// Note: This should be initialized in constructor instead using values in some data file
//
/////////////////////////////
void LocoEngine::findMove()
{

```













```

        {
            0, 0, 0, X, X, 0, 0, X, X, 0, 0, X, X, X, X, X, X } //11 O
        {
            0, 0, 0, X, X, 0, 0, X, X, 0, 0, X, X, X, X, X, X } //12 N
        {
            0, 0, 0, X, X, 0, 0, X, X, 0, 0, X, X, X, X, X, X } //13
        {
            0, 0, 0, X, X, 0, 0, X, X, 0, 0, X, X, X, X, X, X } //14
        {
            0, 0, 0, X, X, 0, 0, X, X, 0, 0, X, X, X, X, X, X } //15
        {
            0, 0, 0, X, X, 0, 0, X, X, 0, 0, X, X, X, X, X, X } //16
        {
            0, 0, 0, X, X, 0, 0, X, X, 0, 0, X, X, X, X, X, X } //17
        {
            0, 0, 0, X, X, 0, 0, X, X, 0, 0, X, X, X, X, X, X } //18
        {
            0, 0, 0, X, X, 0, 0, X, X, 0, 0, X, X, X, X, X, X } //19
        {
            0, X, X, 0, 0, X, X, 0, 0, X, X, X, X, X, X, X, X } //20
        },
    }
    ///////////////////////////////////////////////////
    //
    // Move: 14
    //
    //                                     VELOCITY
    //
    //          0   1   2   3   4   5   6   7   8   9   10  11  12  13  14  15  16
    ///////////////////////////////////////////////////
    {
        0, 0, 0, 0, X, 0, 0, 0, X, 0, 0, X, X, X, X, X, X } // 0
    {
        0, 0, 0, X, X, 0, 0, X, X, 0, 0, X, X, X, X, X, X } // 1
    {
        0, 0, 0, X, X, 0, 0, X, X, 0, 0, X, X, X, X, X, X } // 2
    {
        0, 0, 0, X, X, 0, 0, X, X, 0, 0, X, X, X, X, X, X } // 3
    {
        7, 7, 7, X, X, 7, 7, X, X, 7, 7, X, X, X, X, X, X } // 4
    {
        0, X, X, 9, 9, X, X, 9, X, X, X, X, X, X, X, X, X } // 5 P
    {
        0, 0, 0, X, X, 0, 0, X, X, 0, 0, X, X, X, X, X, X } // 6 O
    {
        0, 0, 0, X, X, 0, 0, X, X, 0, 0, X, X, X, X, X, X } // 7 S
    {
        0, 0, 0, X, X, 0, 0, X, X, 0, 0, X, X, X, X, X, X } // 8 I
    {
        7, 7, 7, X, X, 7, 7, X, X, 0, 0, X, X, X, X, X, X } // 9 T
    {
        0, X, X, 11, X, X, 11, 11, X, X, X, X, X, X, X, X } // 10 I
    {
        0, 0, 0, X, X, 0, 0, X, X, 0, 0, X, X, X, X, X, X } // 11 O
    {
        0, 0, 0, X, X, 0, 0, X, X, 0, 0, X, X, X, X, X, X } // 12 N
    {
        0, 0, 0, X, X, 0, 0, X, X, 0, 0, X, X, X, X, X, X } // 13
    {
        0, 0, 0, X, X, 0, 0, X, X, 0, 0, X, X, X, X, X, X } // 14
    {
        0, 0, 0, X, X, 0, 0, X, X, 0, 0, X, X, X, X, X, X } // 15
    {
        0, 0, 0, X, X, 0, 0, X, X, 0, 0, X, X, X, X, X, X } // 16
    {
        0, 0, 0, X, X, 0, 0, X, X, 0, 0, X, X, X, X, X, X } // 17
    {
        0, 0, 0, X, X, 0, 0, X, X, 0, 0, X, X, X, X, X, X } // 18
    {
        0, 0, 0, X, X, 0, 0, X, X, 0, 0, X, X, X, X, X, X } // 19
    {
        0, X, X, 0, 0, X, X, 0, 0, X, X, X, X, X, X, X, X } // 20
    },
};

dataPtr->move = moveArray[dataPtr->move][position][velocity];

// The following conditionals smooth movement transitions
if (dataPtr->move == 3 || dataPtr->move == 5) {
    dataPtr->move = 1;
}

if (dataPtr->move == 4 || dataPtr->move == 6) {
    dataPtr->move = 2;
}

if (dataPtr->move == 0) {
    if (old_move == 1 || old_move == 2) {
        if (cycle <= delay) {
            dataPtr->move = old_move;
            cycle++;
        }
    }
}

else if (cycle != 0) {
    cycle = 0;
}
*/
cout << setw(6) << "Move:" << setw(2) << move << endl;
*/

```

```
}

///////////////////////////////
//  
// saveOldData  
//  
// Save old angular position data for next loop.  
//  
/////////////////////////////
void LocoEngine::saveOldData()  
{  
    // Save old data  
    old_lh = dataPtr->lh;  
    old_lp = dataPtr->lp;  
    old_lr = dataPtr->lr;  
    old_rh = dataPtr->rh;  
    old_rp = dataPtr->rp;  
    old_rr = dataPtr->rr;  
}
```

THIS PAGE INTENTIONALLY LEFT BLANK

## LIST OF REFERENCES

- [1] Paul M. Fitts, "The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement," *Journal of Experimental Psychology*, Vol. 47, No. 6, pp. 381-391, June 1954.
- [2] D. Meyer, J. Smith, R. Adams, C. Wright, *Attention and Performance*, 13<sup>th</sup> Ed., pp. 173-226, Hillsdale, 1990.
- [3] D. Meyer, J. Smith, R. Adams, and C. Wright, "Optimality in Human Motor Performance: Ideal Control of Rapid Aimed Movements," *Psychological Review*, Vol. 95, No. 3, pp. 340-370, 1988.
- [4] S. Card, W. English, B. Burr, "Evaluation of Mouse, Rate-Controlled Isometric Joystick, Step Keys, and Text Keys for Text Selection on a CRT," *Ergonomics*, Vol. 21, No. 8, pp. 601-613, 1978.
- [5] J. Accot and S. Zhai, "Refining Fitts' Law Models for Bivariate Pointing," *Proceedings of ACM CHI 2003 Conference on Human Factors in Computing Systems*, pp. 193-200, 2003.
- [6] D. Bowman, D. Koller, and L. Hodges, "Travel in Immersive Virtual Environments: An Evaluation of Viewpoint Motion Control Techniques," *Proceedings of Virtual Reality Annual International Symposium (VRAIS)*, 1997.
- [7] L. Arns, *A New Taxonomy for Locomotion in Virtual Environments*. Ph.D. Dissertation, Iowa State University, Ames, Iowa, 2002.  
<http://www.vrac.iastate.edu/~arns/dissertation>, April 2004.
- [8] L. Squire, F. Bloom, N. Spitzer, *Fundamental Neuroscience*, 2<sup>nd</sup> Edition, pp. 425-426, Academic Press, 2003.
- [9] R. Berne, M. Levy, *Principles of Physiology*, 3<sup>rd</sup> Edition, ch. 9, Mosby, 2000.
- [10] Meyer, K., Applewhite, H. & Biocca, F., "A Survey of Position Trackers," *Presence*, Vol. 1, No. 2, pp. 173-200, 1992.
- [11] Eray Unguder, *The Effects of Natural Locomotion on Maneuvering Task Performance in Virtual and Real Worlds*, Master's Thesis, Naval Postgraduate School, Monterey, California, 2001.

- [12] M. Mine, F. Brooks, and C. Sequin, "Moving Objects in Space: Exploiting Proprioception in Virtual-Environment Interaction," *Proceedings of the ACM SIGGRAPH '97 Conference on Computer Graphics*, pp. 19-26, 1997.
- [13] R. Darken, D. Carmein, "The Omni-Directional Treadmill: A Locomotion Device for Virtual Worlds," *Proceedings of UIST '97*, Banff, Canada, pp. 213-221, October 14-17, 1997.
- [14] J. Garbi, "Walking on Virtual Tiles: New Technology Helps Parkinson's and Stroke Patients Walk. Virtual Reality for Aiding People with Movement Disorders." <http://www.cs.technion.ac.il/~baram/tiles.html>. April 2004.
- [15] M. Slater, M. Usoh, and A. Steed, "Taking Steps: The Influence of a Walking Technique on Presence in Virtual Reality," *ACM Transactions on Computer-Human Interaction Computer-Human Interaction*, Vol. 2, No. 3, pp. 201-219, September 1995.
- [16] J. Isdale, "Motion Platforms," VR News, Vol. 9, No. 3, pp. 4-5, April 2000. <http://www.vrnews.com/issuearchive/vrn0903tech.htm>. April 2004.

## **INITIAL DISTRIBUTION LIST**

1. Defense Technical Information Center  
Ft. Belvoir, Virginia
2. Dudley Knox Library  
Monterey, California